Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Avaliando a influência de degradações em imagens nos modelos de aprendizado profundo utilizados em reconhecimento facial

Leandro Dias Carneiro

Dissertação apresentada como requisito parcial para
conclusão do Mestrado em Informática

Orientador
Prof. Dr. Flávio de Barros Vidal

Brasília
2023

Ficha Catalográfica de Teses e Dissertações

Está página existe apenas para indicar onde a ficha catalográfica gerada para dissertações de mestrado e teses de doutorado defendidas na UnB. A Biblioteca Central é responsável pela ficha, mais informações nos sítios:

http://www.bce.unb.br
http://www.bce.unb.br/elaboracao-de-fichas-catalograficas-de-teses-e-dissertacoes

Esta página não deve ser inclusa na versão final do texto.

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Avaliando a influência de degradações em imagens nos modelos de aprendizado profundo utilizados em reconhecimento facial

Leandro Dias Carneiro

Dissertação apresentada como requisito parcial para
conclusão do Mestrado em Informática

Prof. Dr. Flávio de Barros Vidal (Orientador)
Departamento de Ciência da Computação - UnB

Prof. Dr. Hélio Pedrini
Membro Externo - IC/UNICAMP

Prof. Dr. Bruno L. Macchiavello Espinozza
Membro Interno - CIC/IE/UnB

Prof. Dr. Camilo Chang Dorea
Membro Suplente - CIC/IE/UnB

Prof. Dr. Ricardo Pezzuol Jacobi
Coordenador do Programa de Pós-graduação em Informática

Brasília, 21 de Dezembro de 2023

Dedicatória

Dedico este trabalho aos meus pais, que sempre estimularam a busca por objetivos e a luta por conhecimento.

Agradecimentos

Agradeço a Deus, por me dar toda força e resiliência necessárias para conquistar mais um objetivo na vida. A minha extraordinária família, que mesmo longe permanece unida. A minha amada esposa, que compreendeu os momentos de ausência, que incondicionalmente me deu incentivo e suporte durante toda jornada, e ao meu filho, que mesmo sem idade para entender, me dá combustível para alcançar voos maiores.

Resumo

Durante a Persecução Penal, os sistemas de Reconhecimento Facial têm sido cada vez mais utilizados, pois além da acurácia dos sistemas terem aumentado nos últimos anos, observa-se cada vez mais a presença de câmeras nas vias públicas, residências e estabelecimentos comerciais. Atualmente, a maioria dos sistemas comerciais apresenta como resultado uma métrica que representa a similaridade entre duas faces, ou simplesmente uma descrição qualitativa, deixando de lado outras análises a respeito da qualidade e da real utilidade do material utilizado para a comparação. Este trabalho tem como objetivo estimar o impacto que as degradações da imagem causam nos sistemas de reconhecimento facial baseados em aprendizado profundo, a fim de minimizar equívocos cometidos na análise do resultado. Para atingir este objetivo, serão realizadas duas etapas sequencias, sendo a primeira, a criação de uma base de dados e, a segunda, um modelo capaz de identificar a degradação (e a intensidade) presente na imagem. A base de dados será criada a partir de 3 (três) algoritmos de detecção facial, 8 (oito) algoritmos de reconhecimento facial, 14 (quatorze) tipos de degradações com 6 (seis) níveis de intensidade em cada, e 4 (quatro) bases de dados de faces, sendo calculados os escores para as métricas acurácia, *precision* e *recall*. Após a criação da base de dados, será desenvolvido um modelo de aprendizado profundo, capaz de identificar a degradação presente na imagem. Com esta identificação, será possível consultar os resultados da base de dados e estimar a queda de desempenho para as imagens novas. Para as bases de dados de faces analisadas, os modelos de reconhecimento facial tiveram um impacto mínimo de 17%, em média, e um impacto máximo de 43%, em média. Ainda, os modelos treinados na tarefa de detecção de degradação tiveram uma acurácia variando entre 71% e 94%, aproximadamente. Tanto os algoritmos quanto as bases de dados de faces são públicos. O objetivo final do projeto se dá pela identificação dos limites de qualidade necessários para um resultado considerado robusto por parte dos sistemas de reconhecimento facial. Ainda, a criação de um modelo capaz de estimar, com razoável acurácia, o tipo de degradação presente em uma imagem.

Palavras-chave: reconhecimento facial, degradação da imagem da face, qualidade da imagem da face

Abstract

Facial comparison systems have been increasingly used during Criminal Prosecution. Besides the accuracy of the methods has increased considerably in recent years, cameras on public roads and homes also contributed to this advance. Currently, most commercial systems present a metric representing the similarity between two faces or even a qualitative description, leaving aside other analyses regarding the quality and the utility of the material used for comparison. This work aims to estimate the impact that image degradation causes in facial recognition systems based on deep learning. In this way, a database will be created, and then a model capable of identifying the degradation (and intensity) present in the image will be made. The base data will be assembled from 3 (three) facial detection algorithms, 8 (eight) facial recognition algorithms, 14 (fourteen) types of degradation with 6 (six) levels of intensity each, and 4 (four) databases, calculating the scores for the metrics accuracy, *precision* and *recall*. In addition to the database created, a deep learning model will be developed, capable of identifying the manipulation present in the image. With this identification, it will be possible to consult the database results and estimate the drop in performance for new images. For the analyzed face databases, facial recognition models had a minimum impact of 17% on average and a maximum impact of 43% on average. Furthermore, the models trained in the degradation detection task had an accuracy ranging between 71% and 94%, approximately. Both algorithms and databases are public. The project's final objective is to identify the quality limits necessary for a result considered robust by facial recognition systems. Furthermore, it creates a model capable of estimating, with reasonable accuracy, the type of degradation present in an image.

Keywords: face recognition, face image degradation, face image quality

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

A qualidade dos sistemas baseados em visão computacional evoluíram significativamente na última década, incluindo neste contexto, os algoritmos de classificação de imagens, detecção e reconhecimento de objetos e, segmentação de imagens [1]. Esta evolução se deu, em boa parte, devido aos avanços de modelos de aprendizagem profunda (do inglês *deep learning* - DL)[22], principalmente com o advento das Redes Neurais Convolucionais (do inglês *Convolutional Neural Network* - CNN) [23, 24, 25]. Um dos campos de estudo que visivelmente acompanhou esta evolução são os sistemas de reconhecimento facial, os quais vêm sendo objeto de atenção de pesquisadores há algumas décadas devido à sua importância e aplicação na sociedade [1].

O reconhecimento facial, considerado uma das diversas formas de identificação biométrica humana existentes, possui vantagens e desvantagem em relação às outras formas, em especial à impressão digital e ao reconhecimento de íris[26, 1, 27]. Quando realizado de forma automatizada, os sistemas de reconhecimento facial possuem a desvantagem de serem passíveis de interferência ocasionada por artefatos na imagem que prejudicam no seu desempenho, como cosméticos, oclusões, disfarces, condições de luminosidade, baixa resolução, como descrito nos trabalhos de [26, 1].

Entretanto, as muitas vantagens tornam o reconhecimento facial automatizado um dos principais sistemas biométricos utilizados e pode ser visto com certa frequência no cotidiano das pessoas[4]. Neste ponto, se pode considerar que o reconhecimento facial é eficiente para a autenticação e identificação de acesso de pessoas, já que são analisadas as suas características faciais, não necessitando de um especialista, como, por exemplo, no caso de impressões digitais[1, 4]. Além disso, trata-se de uma técnica não-intrusiva, onde o dispositivo de captura pode ser posicionado de maneira mais amigável (ou até mesmo imperceptível para o usuário), sem contato físico, permitindo uma coleta menos constrangedora [1].

Somando-se às duas vantagens anteriores, a captura da imagem exige menos interação

com a pessoa, se comparada com a captura da impressão digital ou da íris. Em alguns casos, a captura pode ser realizada, inclusive, sem a participação ativa do indivíduo, como em sistemas de vigilância em locais públicos, tais como aeroportos e vias públicas, por exemplo [1].

A tecnologia já é também amplamente utilizada na Persecução Penal em diversos países[28]. Em casos onde o único vestígio de um crime é um registro audiovisual (vídeo e/ou imagem), o reconhecimento facial automatizado possui um papel central nas investigações, muitas vezes sendo o elemento norteador do início da investigação[28].

Vale lembrar que no Estado Democrático de Direito brasileiro, o procedimento criminal engloba duas fases de acordo com[29], a saber: a investigação criminal e o processo penal. A investigação criminal é um procedimento preliminar, de caráter administrativo, que busca reunir provas a fim de formar o juízo do representante ministerial acerca da existência de justa causa para o início da ação penal. Já o processo penal é o procedimento principal, de caráter jurisdicional, que termina com um procedimento judicial que resolve se o cidadão acusado deverá ser condenado ou absolvido. Ao conjunto dessas duas fases, dá-se o nome de Persecução Penal[29].

Assim, tanto na fase de investigação criminal quanto no processo penal, os sistemas de reconhecimento facial são utilizados para auxiliar as autoridades durante o processo[28]. Em regra, após a detecção e o reconhecimento de uma face pelo sistema, as imagens são submetidas a análise manual por peritos criminais forenses, onde pode-se suportar ou rejeitar determinada similaridade detectada. Este trabalho forense resulta em um documento oficial, denominado Laudo de Perícia Criminal, e pode gerar uma robusta prova para o Devido Processo Legal, corroborando diretamente na inocência ou na culpabilidade de um suspeito[30].

A tarefa do reconhecimento facial automatizado vem sendo objeto de pesquisa nas últimas décadas, entretanto, nos últimos anos se tem visto de forma constante os níveis de acurácia e desempenho sendo melhorados[27, 31]. Atualmente, após a popularização das CNNs, pode-se considerar que a tarefa de reconhecimento facial é executada de forma bem sucedida, especialmente para imagens obtidas em ambientes favoráveis (ambientes controlados), seguindo os protocolos e/ou melhores práticas para captura das faces sugeridas nas ISO 19194-5 - *Face image data*[32], ICAO-9303 - *Machine Readable Travel Documents*[33] e FISWG (*Facial Identification Scientific Working Group*)[34]. Exemplificando o sucesso do reconhecimento facial, pode-se citar a base de dados LFW[35], a qual é muito utilizado como padrão de desempenho (*benchmark*) para este tipo de aplicação, e hoje em dia é considerado relativamente fácil para diversos algoritmos, os quais atingem escore de acurácia próximo de 100%[26].

Porém, apesar do sucesso no reconhecimento em diversas áreas de aplicação, a tecnologia ainda encontra dificuldades em situações onde o ambiente não é controlado, isto é, onde existem adversidades na captura das imagens. Para aplicações como carros autônomos, vigilância em locais públicos, locais de baixa luminosidade e equipamentos de captura de baixa qualidade, por exemplo, as imagens obtidas nem sempre são o ideal para o correto processamento da aplicação [26, 31]. Diversas pesquisas vêm sendo realizadas com ambiente não controlado e, ao analisar os resultados, verifica-se que os escores obtidos ficam aquém, se comparados com os realizados em ambiente controlado[2].

No trabalho pericial, verificam-se diversos cenários do uso do reconhecimento facial pouco abordados pelos estudos de forma geral. Em regra, a utilização dos sistemas de reconhecimento facial utilizam uma imagem de boa qualidade, com a pessoa na posição canônica, chamada de imagem padrão, e outra imagem com a pessoa em outro cenário, que pode ser em ambiente controlado ou em ambiente não controlado, chamada de imagem questionada[4].

Entretanto, esta prerrogativa não é necessariamente seguida durante uma investigação policial, devido a diversos fatores inerentes à investigação. Desta forma, em diversos casos, ambas as imagens utilizadas no sistema de reconhecimento facial foram obtidas em ambiente não controlado, como de redes sociais, campanas nas vias públicas, mini-câmeras fixadas ao corpo, entre outros casos, impondo aos sistemas de reconhecimento facial um contexto ainda mais difícil de ser analisado[28, 36].

Neste trabalho, a contribuição se dá por um melhor entendimento do impacto no desempenho dos sistemas de reconhecimento facial quando fornecidas imagens degradadas como entrada do sistema. Desta forma, ao inserir degradações nas imagens originais das bases de dados, o experimento busca simular as imagens obtidas em ambiente não controlado, se aproximando do cenário usualmente enfrentado. Assim, será analisado o fluxo de processamento padrão de vários sistemas, variando as imagens de entrada, em relação ao tipo e à intensidade da degradação.

Além disso, o trabalho buscou simular cenários próximos dos encontrados em situações periciais reais, conforme será demonstrado ao longo deste manuscrito. Somando-se a isso, os resultados produzidos pelos experimentos podem ser utilizados como forma de conhecimento produzido para novas pesquisas a partir destes, poupando para novos pesquisadores tempo de desenvolvimento e tempo de execução em *hardware*.

Durante a realização do trabalho, foram analisados e comparados 8(oito) algoritmos de detecção facial (VGG-Face[37], FaceNet[38], OpenFace[39], DeepFace[40], DeepID[41][42], ArcFace[17], Dlib[14] e SFace[18]), 3 (três) algoritmos de reconhecimento facial (Dlib[14], MTCNN[43], RetinaFace[16]), 14 (quatorze) tipos de degradações com 6 (seis) níveis de intensidade para cada tipo de degradação e imagens de 4 (quatro) conjuntos de imagens

públicas. Os resultados do experimento foram analisados a fim de quantificar o impacto das degradações e comparar a robustez dos modelos utilizados.

Após a criação da base de dados, será criado um modelo baseado em aprendizagem profunda que seja capaz de identificar o tipo de degradação presente em uma imagem, e, além disso, a intensidade da degradação detectada. A partir desta identificação, será possível consultar a base de dados gerada na fase anterior e estimar o impacto que essas degradações causariam no desempenho do sistema de reconhecimento facial. Em resumo, ao processar um novo par de imagens, será possível identificar a degradação presente (se houver) e consultar a base de dados gerada, a partir daí, estimar a perda de desempenho do algoritmo de reconhecimento facial.

1.1 Objetivos

1.1.1 Objetivo Geral

Quantificar, comparar, consolidar e estimar automaticamente o impacto ocasionado pelas imagens degradadas da face no desempenho dos sistemas de reconhecimento facial considerados estado da arte, utilizando conjuntos de dados relevantes (*datasets*) para fins de padronização.

1.1.2 Objetivos Específicos

1) Quantificar o impacto na eficácia dos sistemas de reconhecimento facial devido ao uso de imagens degradadas como entrada;

2) Analisar e comparar o desempenho de diversos algoritmos considerados estado da arte;

3) Consolidar os resultados e produzir uma base de dados para consulta; e

4) Identificar automaticamente a degradação de um par de imagens para estimar o impacto no sistema de reconhecimento facial. A identificação usará um modelo de aprendizado profundo treinado para esta tarefa e a estimativa será obtida consultando a base consolidada criada no item anterior.

1.2 Justificativas

Quando utilizados em uma Persecução Penal, os erros de entendimento dos sistemas de reconhecimento facial, podem significar uma prisão ou uma absolvição indevida[36,

28]. Em ambos os casos, o prejuízo para a pessoa ou para a sociedade é irreparável. Contudo, caso os sistemas sejam suportados por estimativas de desempenho considerando a degradação presente nas imagens analisadas, novas conclusões a respeito do resultado podem ser realizadas, e consequentemente, o cometimento de graves equívocos pode ser minimizado.

Este presente estudo se justifica pela necessidade de uma compreensão mais abrangente a respeito dos resultados gerados pelos sistemas de reconhecimento facial. Conforme é esperado, imagens degradadas prejudicam o desempenho dos sistemas, entretanto, qual é, especificamente, a mensuração do grau e a extensão deste prejuízo, é fundamental ser definido.

Os resultados dos experimentos aqui documentados foram consolidados e podem servir de base para novos estudos, poupando principalmente o tempo de execução e o *hardware* utilizado para novos pesquisadores. Conforme demonstrado, a pesquisa demandou alto poder de processamento de servidores com placas gráficas, por um tempo relativamente longo, o que poderia inviabilizar outros pesquisadores com menos recursos disponíveis.

Para automatizar todo o processo, foi desenvolvido uma abordagem para definir um modelo de aprendizado profundo que fosse capaz de identificar o tipo de degradação e a intensidade nas imagens faciais de entrada. Desta forma, com base nos resultados gerados anteriormente, pode-se estimar qual o impacto da degradação presente na imagem no algoritmo de reconhecimento facial com precisão.

Em resumo, foi buscado quantificar o impacto das degradações nos sistemas de reconhecimento facial aqui estudados, bem como produzir uma base de resultados consolidada que pudesse servir de base para outras pesquisas. Ainda, desenvolveu uma abordagem utilizando um modelo de aprendizado profundo para permitir a identificação da degradação e estimar o impacto no algoritmo de reconhecimento facial utilizado.

1.3 Organização da Dissertação

Este trabalho é dividido da seguinte forma: no Capítulo 2 são apresentados conceitos teóricos relacionados à tecnologia utilizada nos sistemas de reconhecimento facial, necessários à compreensão de temas que serão tratados em seguida, no Capítulo 3, são apresentados trabalhos relacionados ao desempenho de sistemas de reconhecimento facial quando utilizadas imagens degradadas como entrada. No Capítulo 4, é descrita a metodologia proposta do projeto. No Capítulo 5 são apresentados os resultados dos experimentos realizados, no Capítulo 6 são apresentadas as conclusões da pesquisa, juntamente com os trabalhos futuros e as publicações realizadas com os dados dos experimentos realiza-

dos neste trabalho. Por fim, no Anexo, são expostos, na íntegra, todos os resultados da pesquisa.

Capítulo 2

Fundamentação Teórica

O objetivo deste capítulo é apresentar os pilares básicos que envolvem um sistema de reconhecimento facial e as definições relativas à imagem, incluindo as degradações e o conceito de qualidade da imagem.

2.1 Sistemas Reconhecimento Facial

Os sistemas de biometria sempre atraíram muita atenção já que são amplamente utilizados como métodos de reconhecimento do ser humano. Existem diversos tipos de sistemas reconhecedores de biometria, tais como reconhecedores de face, íris, voz, palma da mão, assinatura, veias, entre outros[4]. Destes, os sistemas de reconhecimento facial são um dos mais importantes devido a sua grande utilização no nosso dia a dia. Consequentemente, a área vem sendo um campo de pesquisa cada vez mais intenso, sendo notório a evolução da acurácia dos sistemas[4].

Na década de 1960, os pesquisadores Woody Bledsoe, Helen Chan Wolf e Charles Bisson [44, 45, 46, 47] iniciaram o desenvolvimento de sistemas de reconhecimento facial automatizados. Neste experimento, o sistema era semi-automatizado, já que contava com a interação do homem para determinar as coordenadas das características faciais em fotografias antes de serem processadas pelo computador. O sistema precisava que fossem determinadas as coordenadas das características faciais, tais como centro das pupilas, canto interno e externo dos olhos e o pico das viúvas na linha do cabelo. As coordenadas foram usadas para calcular 20(vinte) distâncias, incluindo a largura da boca e dos olhos. Após a demarcação das coordenadas, a fotografia era processada por um computador que comparava automaticamente as distâncias para cada fotografia, calculava a diferença entre as distâncias e retornava uma possível correspondência.

No início da década de 1970, Takeo Kanade[48] apresentou um sistema de reconhecimento facial que identificava as características anatômicas da face, como, por exemplo,

o queixo, e calculava a razão entre essas características de forma totalmente automática. Este pode ser considerado o primeiro sistema de reconhecimento facial totalmente automatizado. Apesar da acurácia do sistema ser considerada baixa para os padrões atuais, o experimento demonstrou que era possível realizar o reconhecimento de forma automática, e atraiu o interesse da comunidade acadêmica e fomentos para as pesquisas na área.

Na década de 90, novos algoritmos surgiram e deram outro salto, elevando o estado da arte da época. Os algoritmos Eigenface[49] e Fisherface[50], surgiram nesta época e eram capazes de extrair informações e agrupar os resultados (faces).

O Eigenface[49], desenvolvido em 1991, por Matthew Turk e Alex Pentland, era capaz de localizar de forma confiável um rosto em uma imagem que contém outros objetos, utilizando um método denominado análise de componentes principais (do inglês *Principal Component Analysis* - PCA). Os Eigenfaces[49], que eram autovetores das faces utilizadas no treinamento, eram determinados com base em características globais e ortogonais em rostos humanos. Um rosto humano é calculado como uma combinação ponderada de um número de Eigenfaces[49]. Posteriormente, para avançar o uso de PCA no reconhecimento facial, os autores definiram características Eigenface[49], incluindo olhos *eigen*, bocas e narizes *eigen*.

Em 1997, houve um aprimoramento no método PCA Eigenface de reconhecimento facial, resultando na produção do Fisherfaces[50], que usava análise discriminante linear (do inglês *Linear Discriminant Analysis* - LDA) para produzir os valores Fisherface. Consequentemente, o LDA Fisherfaces se tornou mais utilizado no reconhecimento facial na época.

Durante a evolução das pesquisas, outros algoritmos e métodos foram obtendo melhores resultados, e nos anos 2010s, o grande destaque era os descritores locais, onde filtros locais eram utilizados para obter representações de características das faces[4]. Entretanto, tais "representações superficiais" eram frágeis quando testadas em ambientes não controlados e em situações de variações faciais mais complexas[4].

Naturalmente o campo evoluiu, até que em 2012, a solução de sistema de reconhecimento facial baseada na rede AlexNet[9], ganhou a competição ImageNet[51] com uma larga margem de vantagem, com a utilização de métodos de aprendizado profundo. Com este advento, em especial das Redes Neurais Convolucionais[23] (do inglês *Convolutional Neural Network* - CNN), múltiplas camadas de unidades de processamento são utilizadas para realizar as transformações e a extração das características de uma imagem. Assim, as redes neurais são capazes de aprender vários níveis de representação que correspondem a diferentes níveis de abstração em relação à imagem original. Atualmente, a tecnologia de aprendizado profundo domina o setor e, basicamente, as pesquisas da área se direcionam em dois sentidos, sendo um com a evolução de novas topologias de redes neurais e outro

com a definição de novas funções de erro [26].

Para realizar a tarefa do reconhecimento facial, existe uma combinação de algoritmos que são responsáveis por tarefas específicas dentro deste contexto geral. Desta forma, são necessárias quatro etapas para realizar este processo, normalmente realizadas em sequência, são elas: a detecção da face, o alinhamento da face, a representação da face e a comparação da face. Cada uma destas etapas corresponde a um algoritmo diferente, de forma que o sistema de reconhecimento facial se dá, como dito, pela combinação entre eles [52, 26, 4].

2.1.1 Fluxo Padrão de um Sistema de Reconhecimento Facial

Atualmente os sistemas de reconhecimento facial, em grande maioria, utilizam aprendizado profundo para executar a tarefa[27]. Para o sistema completar o reconhecimento são necessárias quatro etapas: detecção da face, alinhamento da face, representação da face e comparação da face[52]. Essas etapas se constituem em algoritmos especializados em cada tarefa, de modo que podem ser combinados entre si para compor o sistema de forma completa. A combinação dos algoritmos pode resultar em diferentes resultados de acurácia, assim, o arranjo todo deve ser analisado de forma conjunta, a fim de determinar a acurácia do sistema.

As informações podem ser observadas conforme Figura 2.1.

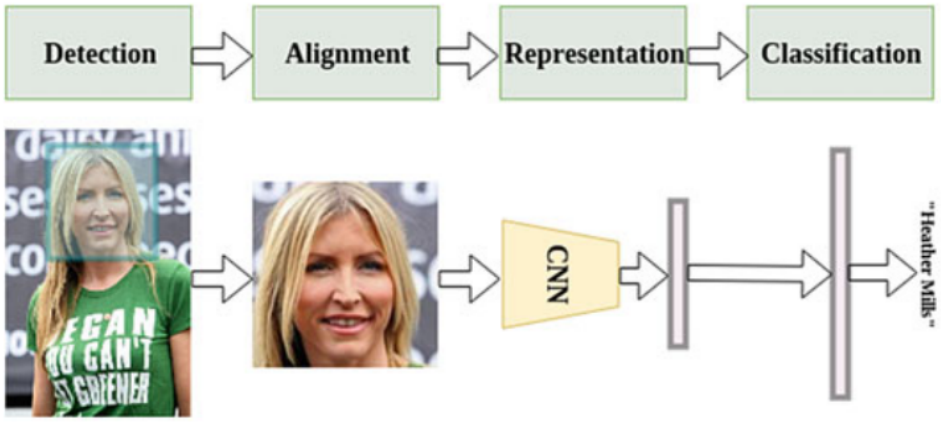


Figura 2.1: Fluxo padrão de um sistema de reconhecimento facial. (Imagem extraída de [1])

Detecção da Face

Normalmente, a detecção da face é o primeiro passo a ser executado. O objetivo é detectar todas as faces existentes em uma imagem[52]. Após a detecção das faces, o algoritmo

retorna as coordenadas das caixas que demarcam as regiões das faces (*bounding boxes*) encontradas, juntamente com um percentual de confiança de cada uma.

Em regra, os algoritmos mais antigos eram testados em uma base que hoje em dia seria considerada pequena, e apesar de os escores de desempenho serem relativamente baixos na época, passava confiança de que o computador pudesse realizar tal tarefa com precisão no futuro. Atualmente, com o uso de aprendizado profundo e grandes bases de dados para treinamento dos algoritmos, a precisão evoluiu significativamente[3, 52].

Através de diversos experimentos, muitos trabalhos destacam que os algoritmos de detecção de faces atuais, baseados em CNNs, são robustos quanto a variações de pose, iluminação e escala. Entretanto, ainda restam diversos outros aspectos que carecem de mais pesquisas, onde os algoritmos encontram mais dificuldades, como, por exemplo, quando trabalham com imagens de baixa resolução e oclusões da face[52].

Alinhamento da Face

Usualmente considerado o segundo passo do reconhecimento facial, o alinhamento da face tem o objetivo de transformar a face detectada no passo anterior para uma visão canônica[52]. Por visão canônica entende-se a face virada para frente, com os olhos alinhados e a angulação da cabeça centralizada. Desta forma, pode-se extrair uma melhor representatividade da face no passo seguinte, já que a posição da face está padronizada.

Cabe mencionar que a pose facial é geralmente representada pelos ângulos de inclinação (*pitch*), guinada (*yaw*) e rotação (*roll*)[4][2]. Ainda, estes critérios de alinhamento facial são definidos pela ISO/IEC 39794-5[53]. Os critérios da pose facial pode ser observada conforme Figura 2.2.

Para o algoritmo realizar o procedimento de alinhamento, é essencial detectar os pontos-chave da face (*facial landmarks*). Os algoritmos podem utilizar pontos faciais diferentes para realizar esta tarefa de alinhamento, sendo que alguns algoritmos utilizam 5 pontos, outros 68 pontos, variando até 194 pontos da face[4][2].

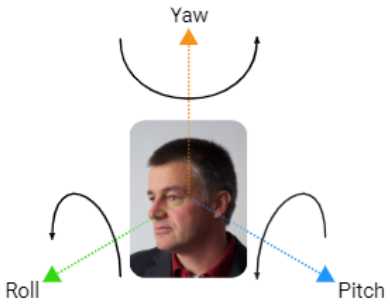


Figura 2.2: Critérios da pose facial observados na ISO/IEC 39794-5. (Imagem extraída de [2])

Representação da Face

O cálculo da representação da face é o ponto central de qualquer sistema de reconhecimento facial, sendo executado após o alinhamento da face (e a geração de uma face canônica)[52], conforme descrito no subitem anterior. O termo representação da face significa um conjunto de valores que representam a face presente em uma imagem. Desta forma, a partir destes valores, é possível calcular a similaridade entre duas faces. Atualmente, para extrair as características e gerar a representação, o uso de aprendizagem profunda tem sido amplamente utilizado pelos principais métodos considerados estados da arte[3][4].

A fim de realizar o cálculo da representação e com a evolução das pesquisas, foram criados algoritmos que utilizam tanto arquiteturas mais genéricas para espinha dorsal do modelo, como mais específicas. Além disso, conforme mencionado, as pesquisas também focaram nas funções de erro, existindo atualmente diversas alternativas criadas especificamente para este problema, onde o objetivo central é diminuir as distâncias intra-classe, ou seja, distâncias entre faces de uma mesma pessoa, e aumentar as distâncias inter-classe, ou seja, distâncias entre faces de pessoas diferentes[3][4].

Em uma CNN utilizada para classificação de imagens, a última camada tem a função de classificação, pois decide a qual classe a imagem pertence. Ou seja, em uma arquitetura de uma rede neural de classificação de imagem, a última camada decide com base nos valores passados pela penúltima camada. Sendo assim, a penúltima camada possui os valores que representam as características da imagem. Portanto, um vetor de representação de uma imagem, neste caso particular, uma face, se dá pela análise dos valores obtidos na penúltima camada de uma rede neural convolucional (para classificação de imagem)[4][27][52]. O vetor de representação da imagem é ilustrado na Figura 2.3.

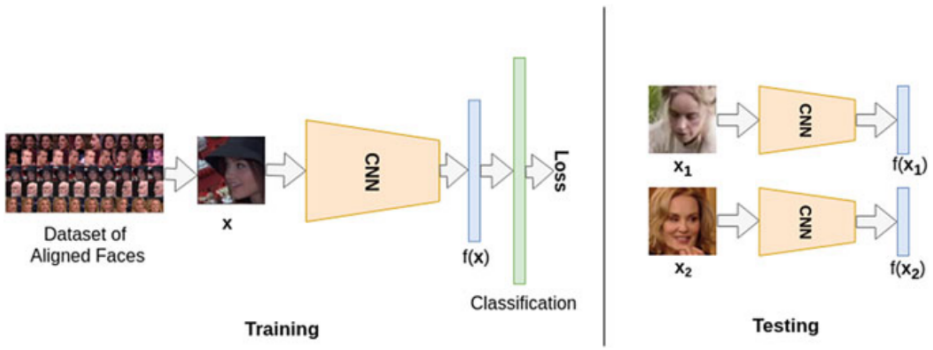


Figura 2.3: Vetor de representação de uma imagem. (Imagem extraída de [2])

Comparação da Face (Cálculo de Similaridade)

O cálculo de similaridade é realizado após as representações das faces serem executadas, e assim, determinar se as faces pertencem à mesma classe (neste contexto, mesma pessoa) ou não. Os extratores de características aprendem durante a fase de treinamento, através das funções de erro. Entretanto, no caso do reconhecimento facial, diferentemente de classificação de objetos, por exemplo, as classes (pessoas) utilizadas no treinamento não estão presentes na fase de teste, daí a necessidade de calcular a semelhança entre as faces através do cálculo de similaridade. Deste forma, o cálculo de similaridade é uma parte essencial em um sistema de reconhecimento facial[52].

Inicialmente o uso da função SOFTMAX foi a escolha mais natural, entretanto, a função não se mostrou robusta o suficiente para este tipo de tarefa[4][54]. Na sequência, o uso da distância euclidiana para o cálculo da similaridade dominou por algum tempo os sistemas de reconhecimento facial[4][52]. Porém, após o advento do treinamento dos algoritmos baseados em margem angular[55], a distância do cosseno passou a dominar as pesquisas mais relevantes[4][55][17]. Atualmente, percebe-se que a maioria dos algoritmos utiliza distância do cosseno para o cálculo de similaridade entre duas representações faciais[3][4]. Neste ponto, cabe informar que as funções de erro mencionadas (e outras) serão explanadas com mais detalhes no Capítulo 4, adiante.

2.2 Tarefas do Reconhecimento Facial

O problema de reconhecimento facial pode ser dividido em duas tarefas, sendo uma a identificação da face e outra a verificação da face. Ambas as tarefas realizam o procedimento de cálculo da representação da face e a comparação com a representação de outra imagem[27]. Em muitos casos, são consideradas tarefas similares, pois utilizam o mesmo fluxo exposto na seção acima: detecção da face, alinhamento da face, representação da face e comparação da face. Entretanto, o objetivo final das tarefas difere[27], conforme detalhado a seguir.

2.2.1 Identificação da Face

Identificação da face significa que dado uma galeria (um grupo de imagens contendo faces) e uma imagem questionada (uma imagem específica contendo uma face) são submetidas a um sistema de reconhecimento facial, o qual deve indicar a imagem pertencente à galeria que mais se assemelha com a imagem questionada[27][4]. Para realizar esta tarefa, é necessário realizar o cálculo de similaridade entre a representação da imagem questionada e todas as representações das imagens pertencentes à galeria, e por fim, selecionar a

imagem com o maior grau de similaridade (ou menor distância facial). Esta tarefa é denominada 1-N ("um-para-muitos") *face matching* em alguns trabalhos[52].

2.2.2 Verificação da Face

Na verificação da face, o objetivo é determinar se duas imagens específicas correspondem à mesma pessoa. Este processo é realizado comparando as representações entre as duas imagens e verificando o grau de similaridade a partir de um limiar[4]. Caso o valor da distância facial esteja abaixo do limiar, as imagens são consideradas da mesma classe, ou seja, da mesma pessoa. Caso a distância fique acima, são consideradas pessoas diferentes. Esta tarefa é denominada 1-1 ("um-para-um") *face matching* em alguns trabalhos[52].

2.3 Aprendizagem Profunda

O uso de aprendizagem profunda é dominante no uso de sistemas de reconhecimento facial atualmente, pois os sistemas conseguem aproveitar a arquitetura das redes neurais profundas para aprender as representações da face de forma discriminativa[3]. A acurácia dos sistemas cresceu, de forma contundente, após a adoção das técnicas e as constantes pesquisas em algoritmos de aprendizagem profunda. O nível de acurácia atualmente obtido pelos sistemas é suficiente para a adoção dos algoritmos em diversas aplicações no mundo real[4].

A seguir, será exposto alguns destes recursos utilizados nas redes neurais profundas mais utilizadas.

2.3.1 Redes Neurais Convolucionais

Uma Rede Neural Convolucional (do inglês *Convolutional Neural Network* - CNN)[23], um tipo de rede neural dentro do segmento aprendizagem profunda, possui notável eficácia em processamento de imagens, sendo utilizada em diversas aplicações de visão computacional, incluindo a tarefa de reconhecimento facial. Neste sentido, [3] analisou 171 contribuições recentes para a área, classificando as técnicas utilizadas no reconhecimento facial, conforme Figura 2.4. Neste gráfico, pode-se observar a predominância do uso de CNNs nas contribuições pesquisadas.

Uma CNN possui a capacidade de processar a informação por sucessivas camadas, realizando diversas extrações de características e transformações, aprendendo múltiplos níveis de representação a partir da imagem original. Desta forma, as CNNs são capazes de aprender em suas camadas mais iniciais as características menos abstratas, tais como

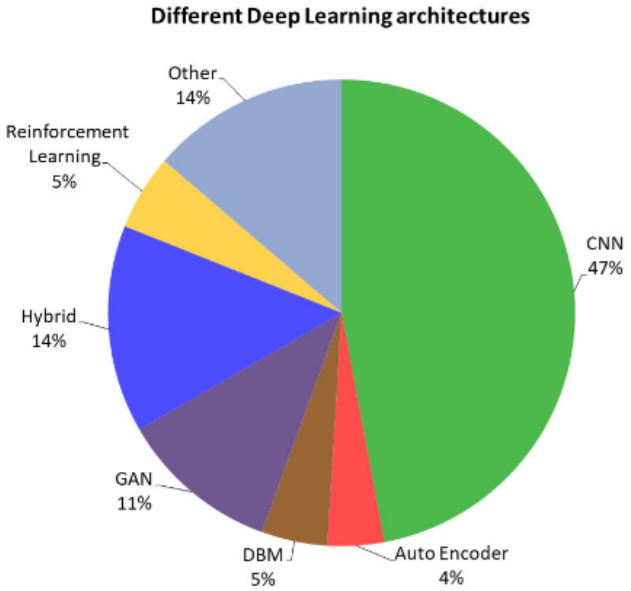


Figura 2.4: Técnicas de aprendizagem profunda para o reconhecimento. (Imagem retirada de [3])

linhas e contornos, e em suas camadas mais profundas características mais semânticas, como reconhecer uma face ou uma pelagem[4].

Conforme exibido na Figura 2.5, a primeira camada da CNN extrai características mais concretas, tais como linhas e contornos, a segunda camada aprende características de textura mais complexas. Já na terceira camada, algumas características abstratas simples já começam a aparecer, tais como nariz e olhos. Na última camada, a rede já é capaz de responder problemas mais complexos, pois já se observa características como sorriso, expressão facial, cor dos olhos, entre outros[4].

2.3.2 Espinha Dorsal dos Modelos de Reconhecimento Facial

As arquiteturas utilizadas nos sistemas de reconhecimento facial, em regra, são aquelas utilizadas para tarefas de classificação de imagens. Conforme explanado, extrai-se a última camada (camada de classificação) e utiliza-se a penúltima camada como saída da rede, representando assim, um vetor de características da imagem[3][4]. Assim, em ordem cronológica, será apresentado as principais arquiteturas utilizadas como espinha dorsal de um sistema de reconhecimento facial.

A arquitetura AlexNet[9] revolucionou o mundo em 2012, quando foi lançada. A arquitetura contava com cinco camadas convolucionais e três camadas densas, além disso, a arquitetura integrou diversas técnicas de aprendizagem profunda, como *dropout*[11], *data augmentation*[13] e a função de ativação ReLU[56] (*Rectified Linear Unit*). A arquitetura

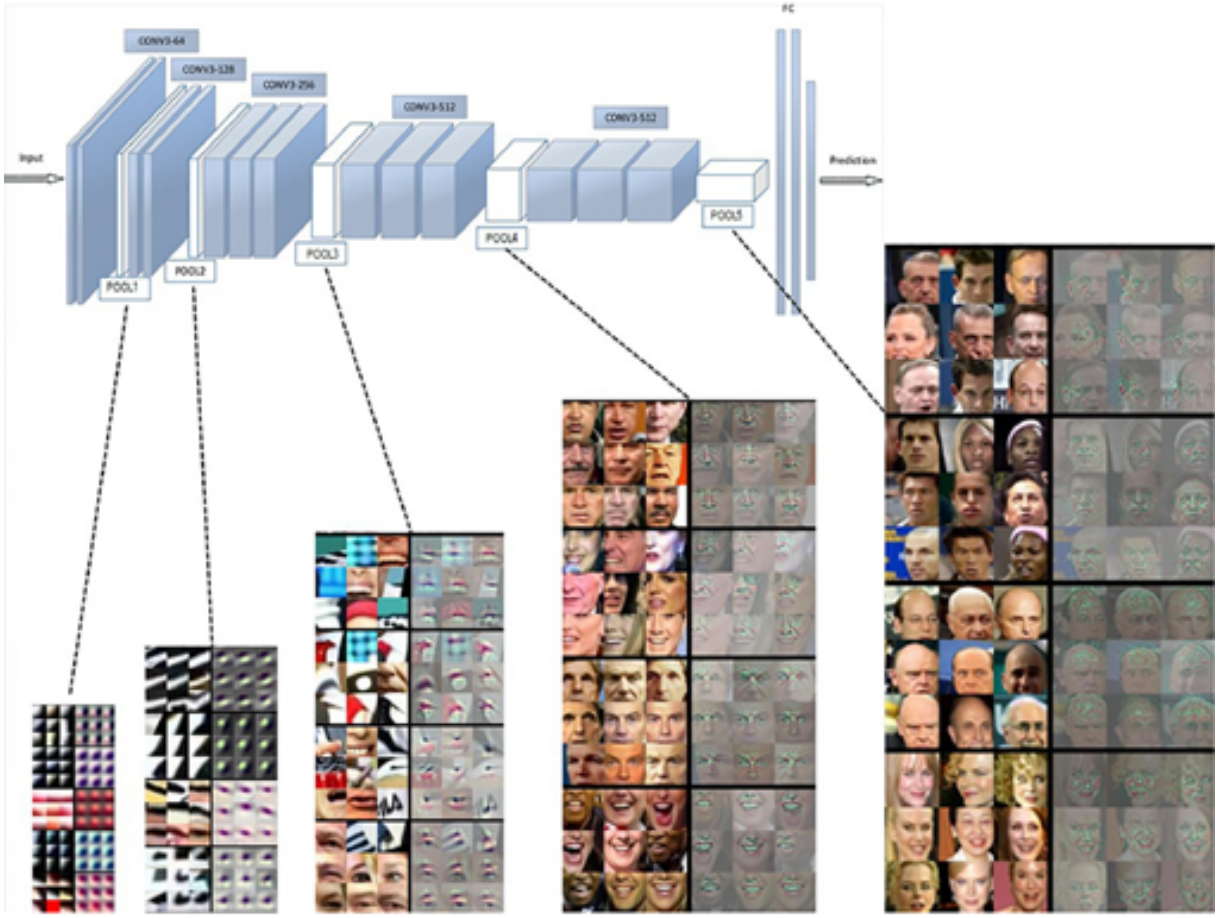


Figura 2.5: Aprendizado da representação em diversas camadas: Modelos de aprendizagem profunda aprendem elementos primários nas camadas mais superficiais e elementos semânticos nas camadas mais profundas. (Imagem retirada de [4])

surgiu durante a competição ImageNet[51] e atingiu o estado da arte da época, obtendo alta acurácia e superando em grande vantagem todas as demais propostas[4].

Em 2014 surgiu a arquitetura VGGNet[10], criada pelo grupo VGG (*Visual Geometry Group*) da Universidade de Oxford, que se caracterizou pela utilização de filtros de convolução de dimensão 3x3, considerado pequeno. A rede também dobrava o número de mapas de características (do inglês *features maps*) a cada camada de *pooling* de 2x2, e aumentou a profundidade da rede, para 16 ou 19 camadas. Este aumento na quantidade de mapas de características e de camadas da rede neural permitiram a rede ter mais flexibilidade e aprender mais características para realizar a classificação da imagem[4].

A rede GoogleNet[8] foi uma arquitetura desenvolvida pela empresa Google, surgiu em 2015, e era composta de 22 camadas de profundidade. Além do incremento no número de camadas, a rede trouxe outras inovações para o campo, já que introduziu o módulo *inception*. Assim, a rede podia performar diversas convoluções simultaneamente, com filtros de tamanhos diferentes (1x1, 3x3 e 5x5), e concatenava os mapas de características para passar como entrada para a próxima camada[8]. O objetivo da rede era aprender, a partir da mesma entrada, mapas de características com filtros de tamanhos diferentes[4].

Em 2016, com o advento da rede ResNet[7] (*Residual Network*) e a implantação das *skip connections*, estrutura que permite as camadas mais profundas aprenderem características diretamente da imagem original e não apenas dos mapas de características das camadas anteriores, foi novamente obtido um incremento na acurácia dos sistemas. A ResNet, foi desenvolvida pela empresa Microsoft, e possui diferentes tamanhos sugestivos (ResNet-18, ResNet-34, ResNet-50, ResNet-101, ResNet-152, entre outros). Além de utilizar diversas técnicas, tais como *dropout* e *batch normalization*, a principal inovação da ResNet se deu pela estrutura *skip-connection*, a qual tornou possível treinar uma rede neural de 152 camadas, já que permitiu resolver o problema de desaparecimento do gradiente (do inglês *vanish gradient*), que impactava nas redes neurais que possuíam muitas camadas[7][4]. Além disso, a *skip-connection* possui um processamento bastante simples, enquanto nas redes neurais anteriores, as camadas eram sequenciais, ou seja, a saída de uma camada era a entrada da seguinte, na ResNet a saída de uma camada é a entrada de 2 (duas) ou 3 (três) camadas seguintes[4]. Apesar da simplicidade de funcionamento, as *skip-connections* representaram um grande salto de acurácia e estabeleceu um novo estado da arte na sua época, pois permitia as camadas mais profundas aprender diretamente a partir das informações brutas da mesma forma que as camadas mais superficiais[4][3].

A Figura 2.6 ilustra uma comparação entre uma Resnet de 34 camadas, uma rede neural sequencial de 34 camadas e uma VGGNet de 19 camadas.

Em 2017, a rede SENet[6] apresentou a estrutura denominada *Squeeze-and-Excitation*. Esta estrutura pode ser integrada com arquiteturas modernas, como, por exemplo, a pró-

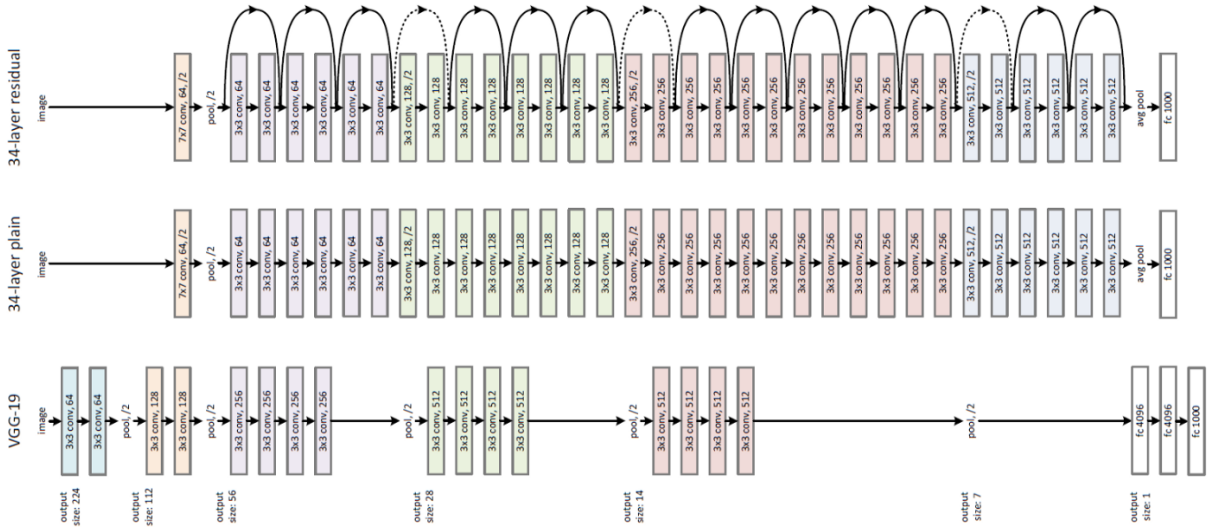


Figura 2.6: ResNet-34, CNN sequencial de 34 camadas e VGGNet-19. (Imagem extraída de [5]

pria ResNet, melhorando sua capacidade de representação da informação com um custo de processamento muito baixo[6]. O bloco *Squeeze-and-Excitation* tem a função de "recalibrar" os mapas de características, pois modela explicitamente as interdependências entre os mapas[4][1][6]. Em resumo, a estrutura adiciona parâmetros a cada mapa de características do bloco CNN, permitindo a rede neural ajustar os pesos de cada mapa. Desta forma, ao passo que nas redes neurais anteriores os mapas de características possuíam o mesmo peso (ou mesma importância), a SENet implanta um mecanismo que permite estabelecer pesos para cada mapa, permitindo que cada mapa de característica tenha um valor individual de relevância. Através de experimentos[6], uma rede ResNet-50 com adição de blocos SENet apresentou a mesma acurácia de uma ResNet-101, a um custo computacional adicional de 1%[4]. Na Figura 2.7, são apresentadas as principais estruturas das redes citadas.

2.4 Elementos Fundamentais em Aprendizagem Profundo

Diversos elementos técnicos contribuiram para melhorar o desempenho das redes neurais profundas. A seguir, descrevemos alguns deles.

2.4.1 *Dropout*

Um problema que impacta no desempenho dos modelos de aprendizagem profunda é o sobreajuste (do inglês *overfitting*). O termo sobreajuste é utilizado para descrever quando

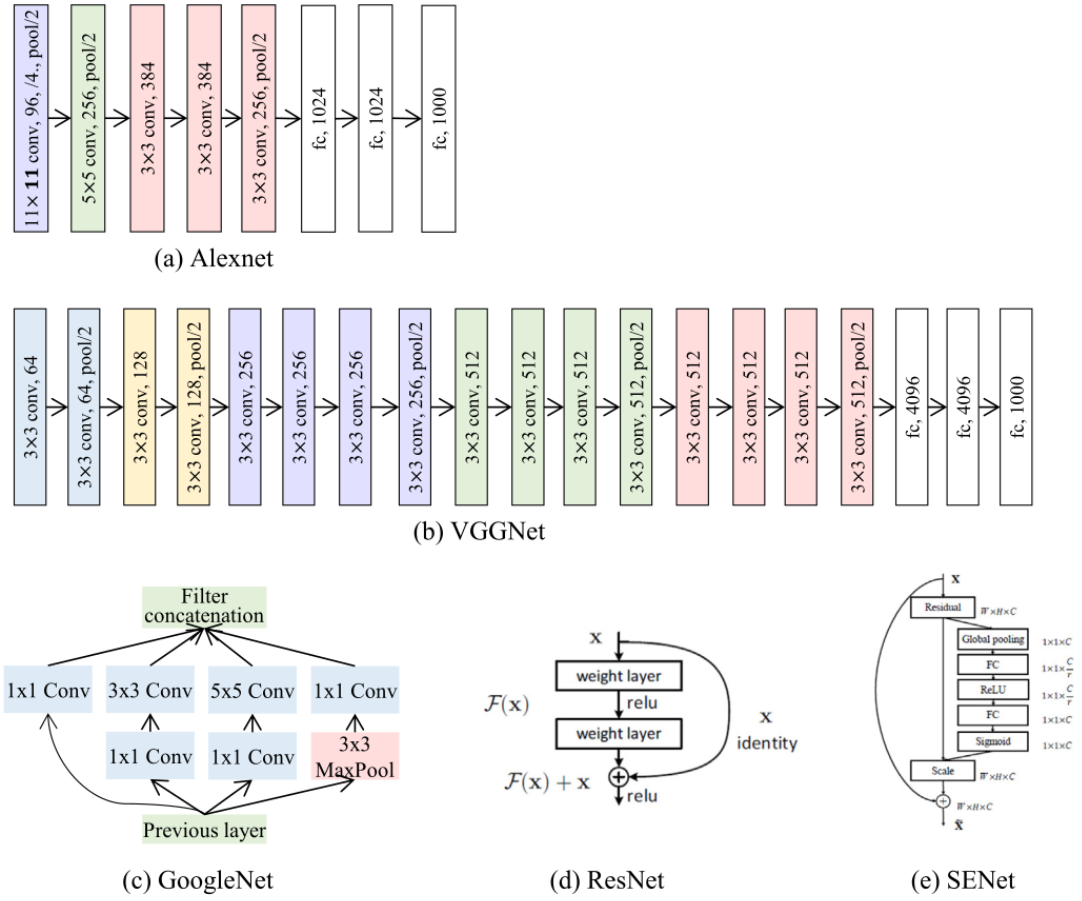


Figura 2.7: Evolução das arquiteturas: (a) Alexnet, (b) VGGNet (Oxford), (c) GoogleNet (Google), (d) ResNet (Microsoft) e (e)SENet. (Imagens extraídas de [6][7][8][9][10]).

um modelo se ajusta muito bem ao conjunto de dados de treinamento, mas se mostra ineficaz para prever novos resultados. Para remediar este problema, em 2014, foi apresentado a técnica denominada *Dropout*[11]. A ideia principal é descartar unidades (neurônios artificiais) aleatoriamente (junto com suas conexões) da rede neural durante o treinamento. Isso evita que as unidades se adaptem demais. Esta solução reduz significativamente o sobreajuste e oferece grandes melhorias em relação a outros métodos de regularização. Além disso, o *Dropout* melhorou o desempenho de redes neurais em diversas tarefas de aprendizado supervisionado, obtendo resultados mais robustos nos testes realizados pelos autores do artigo[11].

2.4.2 *Data Augmentation*

Outra forma de evitar o problema de *Overfitting*, mencionado no subitem anterior, é utilizar uma base de imagens para treinamento extremamente grande e variada. Infelizmente, muitas áreas de conhecimento não possuem acesso a esta quantidade de imagens, ou o custo de aquisição tornaria inviável o projeto, nestes casos, a técnica denominada *Data Augmentation*[13] pode ser uma solução. *Data Augmentation* trata o problema de dados limitados para treinamento já que engloba um conjunto de técnicas que aprimoram o tamanho e a qualidade dos conjuntos de dados de treinamento para que os modelos possam aprender a partir de uma variedade maior de informações. Os algoritmos utilizados em imagens aplicam transformações geométricas, aumentos de espaço de cores, filtros de kernel, imagens de mistura, exclusão e oclusão de pixels aleatoriamente, GANs[57] (*Generative Adversarial Network*), entre outros recursos[13].

2.4.3 *Batch Normalization*

Uma técnica em destaque que contribuiu para o avanço da tecnologia foi o *Batch Normalization*[58], apresentada em 2015. Em regra, para um treinamento mais estável, era necessária uma taxa de aprendizagem (do inglês *learning rate*) bem baixa, e uma inicialização cuidadosa dos parâmetros, o que tornava mais difícil o treinamento dos modelos. A técnica *Batch Normalization* torna a normalização uma parte da arquitetura do modelo, e realiza a normalização a cada mini-lote (*mini-batch*) de treinamento. Esta normalização em mini-lotes permite usar taxas de aprendizagem muito mais altas e uma inicialização menos propícia a erros. Durante os experimentos, foi observado que em alguns casos, o *Dropout* poderia ser até eliminado. Os autores afirmam que, quando a técnica é aplicada a um modelo de classificação de imagens de última geração, pode atingir a mesma precisão com 14 (quatorze) vezes menos etapas de treinamento, e superar o modelo original por uma margem significativa[58].

2.4.4 *Rectified Linear Units* (ReLU)

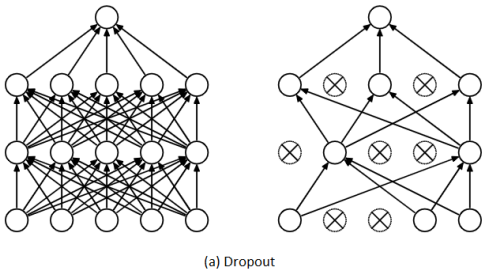
A função de ativação ReLU[59, 12, 56] possui, em geral, desempenho muito superior durante o treinamento em relação às demais funções de ativação observadas, tais como Sigmoid e Tangente Hiperbólica[54], e por isso, é extremamente utilizada na construção de redes neurais.

De forma simples, a função ReLU é linear para os valores positivos e zero para os valores negativos. Ou seja, para os valores negativos, passará o valor zero, e para os valores positivos, uma função linear. A função ReLU é dada pela Equação 2.1, a seguir:

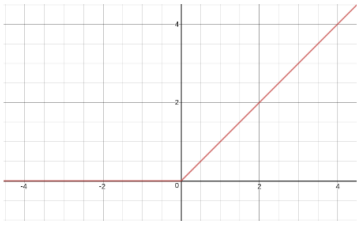
$$f(x) = x^+ = \max(0, x) = \begin{cases} x & \text{if } x > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{2.1}$$

onde x representa o valor de entrada da função.

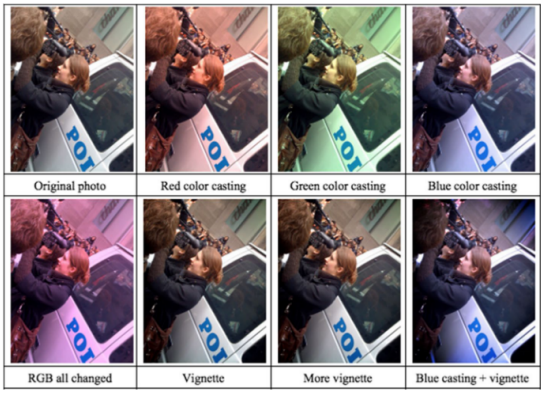
Na Figura 2.8, são apresentadas as técnicas descritas acima.



(a) Dropout



(b) Função ReLU



(c) Data Augmentation (transformações na cor)

Figura 2.8: Evoluções importantes: (a) *Dropout*, (b) Função de ativação ReLU e (c) *Data Augmentation*. (Imagens extraídas de [11][12][13])

2.5 Algoritmos de Detecção Facial

Em regra, os algoritmos de detecção facial são os primeiros a serem executados durante o fluxo de um sistema de reconhecimento facial. Eles são responsáveis por detectar a face e determinar as coordenadas da região de interesse na imagem. A partir desta etapa, a região de interesse pode ser alinhada e, em seguida, ter seu vetor de representação gerado.

A seguir, serão descritos, os modelos de detecção facial utilizados neste trabalho: Dlib[14], MTCNN[43] e RetinaFace[16].

2.5.1 Dlib

A biblioteca Dlib[14], mencionada anteriormente, possui diversas tarefas embutidas dentro dela. A biblioteca permite detectar diretamente uma (ou várias) face(s) a partir de uma imagem. A biblioteca pode retornar as coordenadas das faces encontradas, as coordenadas dos pontos-chave das faces, realizar o alinhamento da face, verificar se a face se encontra frontal, entre outras tarefas. Entretanto, apesar da versatilidade, o algoritmo de detecção facial Dlib não é considerado a melhor opção nesta seara, havendo outras alternativas com maior acurácia[15]. A Figura 2.9 ilustra a detecção dos pontos-chaves de uma face.

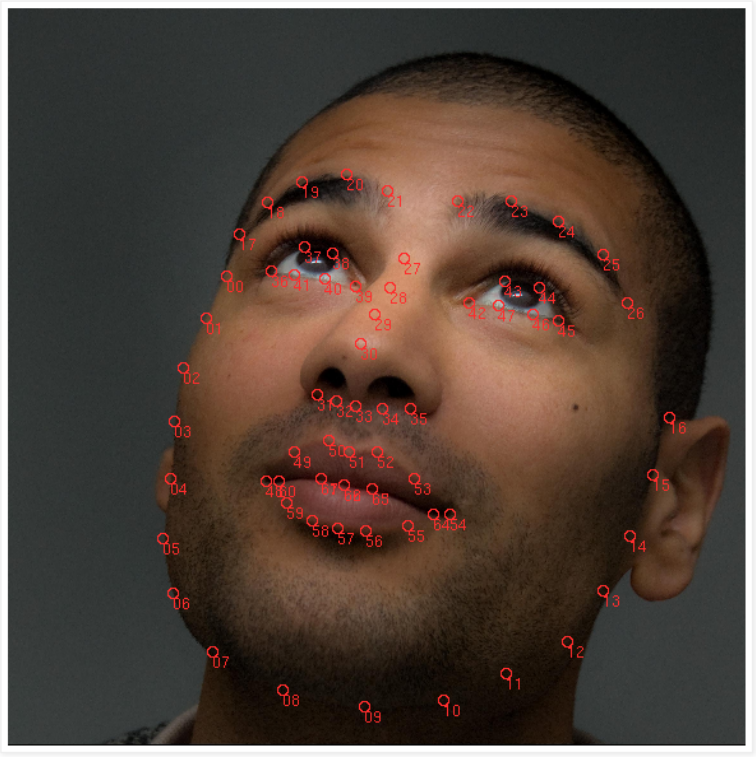


Figura 2.9: Dlib: 64 pontos-chaves. (Imagem extraída de [14])

2.5.2 MTCNN

O algoritmo de detecção facial MTCNN[43] (Multi-Task Cascaded Convolutional Networks) é um dos mais utilizados devido à sua robustez e acurácia elevada[15]. O algoritmo é baseado em redes CNN e é capaz de detectar a face, os pontos-chaves e realizar o alinhamento facial. Entretanto, um ponto considerado negativo deste algoritmo é seu tempo de exe-

cução, já que devido a sua estrutura, é considerado um algoritmo mais complexo, sendo superado no tempo de resposta por outras alternativas[15].

Conforme exposto, a acurácia elevada do MTCNN se deve a sua estrutura mais moderna e complexa. O detector conta com três modelos de CNN para realizar o processamento, são elas: P-Net, R-Net e O-Net[43].

P-Net (Proposal Network): responsável por procurar faces em mini-janelas de tamanhos 12x12. O objetivo deste modelo é produzir resultados de forma rápida, e, consequentemente, possui uma taxa alta de falsos-positivos;

R-Net (Refine Network): Todas as janelas (possíveis faces) indicadas na P-Net são entradas na R-Net. Assim, a R-Net rejeita ou aceita a janela. Na prática, esta rede possui uma alta taxa de rejeição de janelas. Nota-se que a estrutura deste modelo é mais profunda que do modelo P-Net.

O-Net (Output Network): responsável por retornar as caixas delimitadoras da área da face e os pontos-chave da face.

A arquitetura da MTCNN pode ser observada na Figura 2.10.

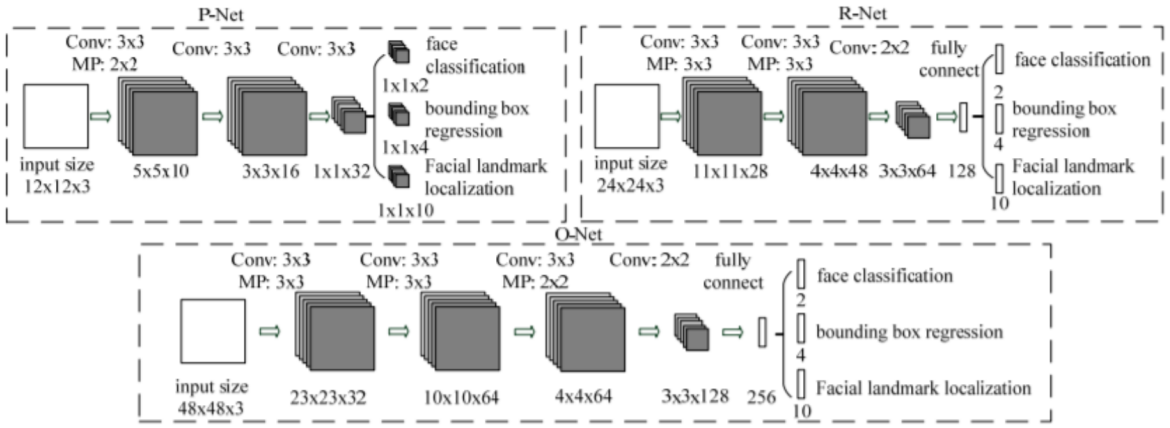


Figura 2.10: Arquitetura MTCNN: P-Net, R-Net e O-Net. (Imagem extraída de [15])

2.5.3 Retinaface

RetinaFace[16] é um algoritmo de detecção facial considerado estado da arte devido ao alto desempenho apresentado[15]. O modelo de detecção facial foi proposto pelo grupo InsightFace[60] em 2019, e, quando utilizada em conjunto com o modelo de reconhecimento facial ArcFace, forma um robusto fluxo para um sistema de reconhecimento facial[15].

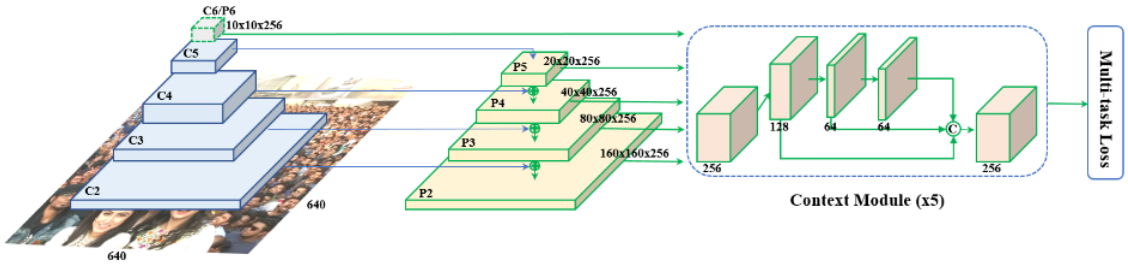


Figura 2.11: Arquitetura RetinaFace. (Imagem extraída de [16])

O modelo proposto é baseado em pirâmides com módulos independentes. Ainda, o modelo possui uma função de erro multitarefa[16]. A arquitetura proposta pode ser observada na Figura 2.11.

Conforme os experimentos apresentados por [16], a combinação RetinaFace + ArcFace apresentou uma melhora na acurácia em relação à combinação MTCNN + ArcFace para as bases de dados utilizadas: LFW, CFP-FP e AgeDB-30. A seguir, na Figura 2.12, pode-se verificar a alta acurácia do algoritmo na detecção das faces.

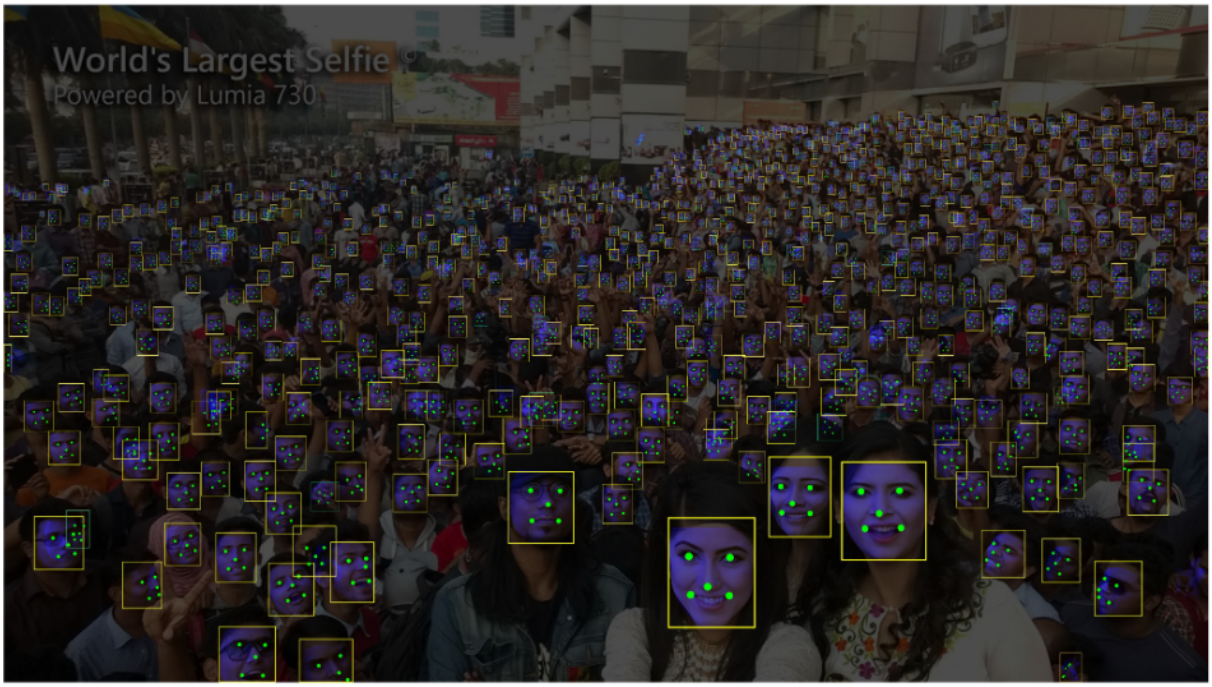


Figura 2.12: RetinaFace encontrou cerca de 900 faces na imagem com total de 1.151 pessoas. (Imagem extraída de [16])

2.6 Algoritmos de Reconhecimento Facial utilizados

O objetivo desta seção é descrever os modelos de reconhecimento facial utilizados durante os experimentos. Considerando o fluxo de um sistema de reconhecimento facial apresentado em 2.1.1, estes modelos são responsáveis por gerar o vetor de representações da face (do inglês *face representation* ou *feature vector*). A seguir, serão descritos, em ordem cronológica, os modelos utilizados neste trabalho: DeepFace da empresa Facebook, DeepID da Universidade de Hong Kong, Dlib da biblioteca multi-tarefa Dlib, FaceNet da empresa Google, VGG-Face da Universidade de Oxford (Inglaterra), OpenFace da Universidade Carnegie Mellon (Estados Unidos), ArcFace da Universidade Imperial de Londres e SFace do instituto Fraunhofer (Alemanha).

2.6.1 DeepFace

Em 2014, os pesquisadores da empresa Facebook anunciaram um modelo de reconhecimento facial, denominado DeepFace[40], o qual atingiu uma acurácia muito semelhante com o obtido pelo olho humano. Enquanto o modelo podia atingir 97,35% de acurácia quando utilizando a base de dados LFW[35], o homem atingia 97,53%. O modelo podia atingir, inclusive, acurácia maior que um homem médio em alguns casos específicos[15].

O modelo contava com uma CNN de 8 (oito) camadas onde cada camada era nomeada de acordo com uma letra e um número, sendo que a letra significava a função da camada e o número a sequência na arquitetura. Desta forma, na Figura 2.13, pode-se observar: C significa *Convolutional Layer*; M significa *Max Pooling Layer*; L, *locally connected layer*; e F, *fully connected layer*[40].

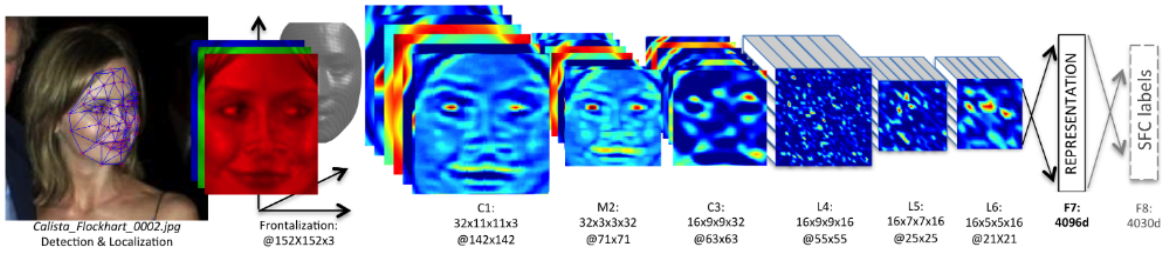


Figura 2.13: Modelo DeepFace. Imagem extraída de [15]

O modelo foi treinado com o *dataset* SFC[61], o qual contém 4,4 milhões de imagens e 4.030 pessoas diferentes. Este é o número de neurônios na última camada (4.030 pessoas)[40], conforme a Figura 2.13.

Ainda conforme a Figura mencionada, observa-se que a última camada (F8) está opaca, isso de dá porque esta camada será excluída e, para o cálculo da representação da face será

utilizado os valores da camada anterior, no caso a penúltima. Esses valores são chamados de vetor de características (do inglês *feature vector*)[40].

O treinamento foi realizado com o uso da função de erro denominada *Cross Entropy Loss*, que, assim como as demais funções de erro analisadas, buscava aproximar as imagens da mesma pessoa (intra-class) e distanciar as de pessoas diferentes (inter-class). Importante informar que o sistema utilizava a distância euclidiana como métrica para o cálculo da similaridade e a função de ativação ReLU para as camadas intermediárias do modelo[15].

Após análise, pode-se observar que o modelo possuía um número relativamente pequeno de camadas, entretanto sua quantidade de parâmetros era considerada grande para a época. Enquanto que o DeepFace possuía 137 milhões de parâmetros, FaceNet (Google) e OpenFace possuíam 22 milhões e 3,7 milhões, respectivamente. Nesta comparação, o modelo VGG-Face possuía uma quantidade de parâmetros compatível, com cerca de 145 milhões[15].

2.6.2 DeepID

Desenvolvido em 2014, a primeira e a segunda versão do modelo possuem muitas similaridades e algumas diferenças. A primeira versão contava com a entrada de imagens em escala de cinza (um canal monocromático), com dimensões de 39x31 pixels. Já a segunda versão permitia entrada de imagens coloridas (três canais), dimensões de 55x47 pixels e era nomeada DeepID2[15].

O modelo contava com quatro camadas de convolução, seguidas de camadas *max-pooling*. Ao final, possuía uma camada *fully-connected*. A rede era treinada com uma camada final de classificação SOFTMAX, semelhante ao treinamento de classificação de imagens, onde existem as classes definidas. Após o treinamento, a camada SOFTMAX era retirada e a saída da rede se dava pela penúltima camada da etapa de treinamento. Esta nova saída gerava um vetor unidimensional com 160 valores e representava as características da face[41].

Uma característica relevante desta arquitetura é que a 3ª camada de convolução está conectada à 4ª camada de convolução e a camada totalmente conectada. A 4ª camada de convolução está conectada à camada totalmente conectada. Desta forma, percebe-se que a camada totalmente conectada recebe como entrada as informações das camadas de convolução 3 (três) e 4 (quatro) e gera um vetor unidimensional de 160 valores[15][41]. Este vetor é a representação da face e pode ser comparado com outro vetor para calcular o grau de similaridade entre duas faces.

A arquitetura do modelo pode ser observada na Figura 2.14.

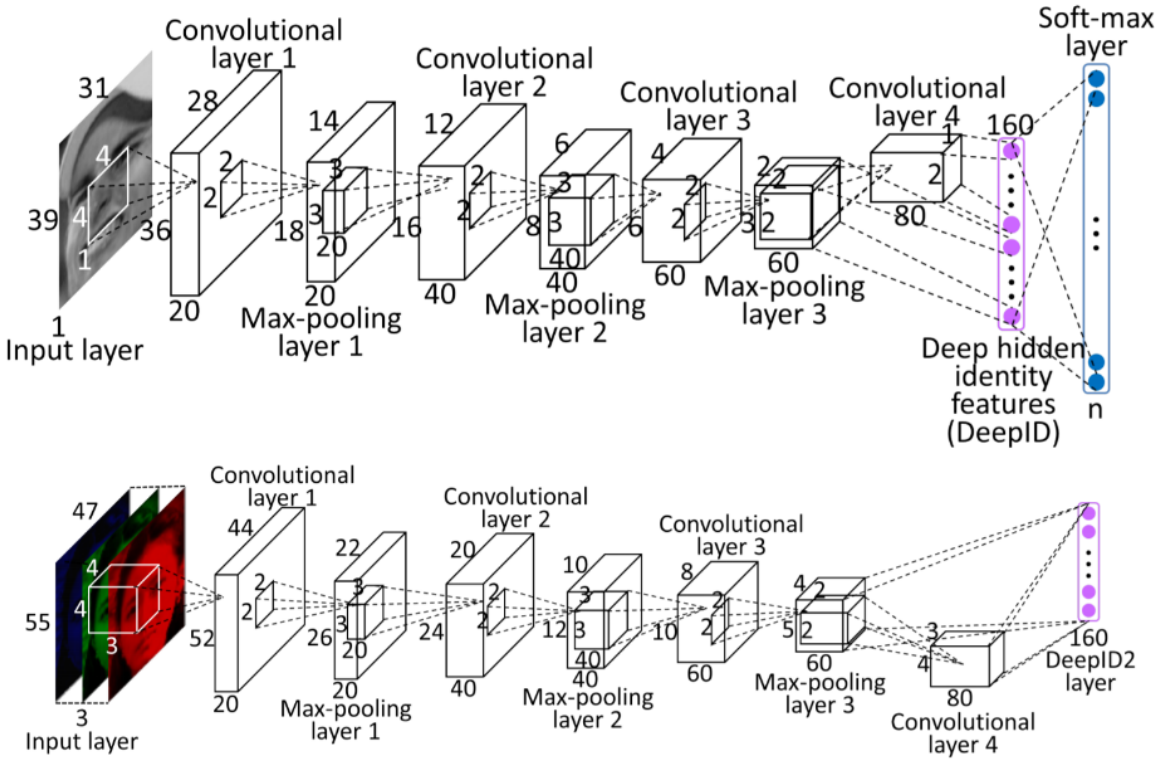


Figura 2.14: Modelo DeepIP, versões 1 e 2. (Imagem extraída de [15]

Este modelo pode ser considerado pequeno e leve, e portanto uma opção a ser considerada para uso em aplicações em tempo real ou que necessitem de muita velocidade[15].

2.6.3 Dlib

A biblioteca multitarefa Dlib[14] apresentou em 2015 um algoritmo de reconhecimento facial flexível e produtivo, já que possuía diversos métodos prontos para o uso, como detecção facial, alinhamento facial, detecção de pontos-chave da face e o cálculo da representação da face[15].

A arquitetura do modelo Dlib para representação da face foi inspirado na ResNet-34[7]. O autor da biblioteca, Davis E. King[62], modificou a estrutura padrão da rede ResNet-34 e retirou algumas camadas, restando 29 ao final[15]. Ainda, a rede foi retreinada e a saída gerava um vetor unidimensional de 128 posições. Este vetor era a representação da face.

O treinamento da rede se deu em diversas bases de dados, incluindo FaceScrub [63] e VGGFace2[64], totalizando mais de 3 milhões de exemplares de faces. O teste de acurácia foi realizado na base de dados LFW[35] e atingiu a impressionante marca de 99,38% de acurácia[14]. Este escore é superior ao atingido pelo olho humano, o qual era de 97,53 e compatível com os algoritmos considerados estado da arte. Ainda, o autor propõe o uso da distância euclidiana para inferir o grau de similaridade entre as faces[15].

Por fim, vale mencionar que a biblioteca Dlib se propõe a realizar o fluxo completo de um sistema de reconhecimento facial: detecção, alinhamento, representação e comparação da face[15].

2.6.4 FaceNet

A rede FaceNet[38] foi criada pela empresa Google em 2015, através de uma base de imagens de cerca de 200 milhões de imagens e 8 milhões de pessoas diferentes.

O modelo calcula a representação das faces e gera um vetor de característica de 128 valores, denominado *face embedding*, que representa as características da face. Assim, é possível mapear as imagens no espaço euclidiano, onde o cálculo da similaridade se dá, naturalmente, pela distância euclidiana entre as faces[15].

Ainda, o modelo utiliza a estrutura de bloco Inception[65], que se caracteriza pelo cálculo de convoluções com filtros de diversos tamanhos simultaneamente. Para o cálculo do erro, é utilizado a função de erro denominada *triplet-loss*, que faz utilização de uma face âncora, uma face positiva e uma face negativa, e o objetivo é aproximar a distância entre a âncora e a face positiva e afastar entre a âncora e a face negativa[15].

2.6.5 VGG-Face

O modelo VGG-Face[37] foi criado em 2015, pelo Visual Geometry Group do Departamento de Engenharia da Universidade de Oxford. Originalmente o modelo foi denominado Deep Face, entretanto, como existia outro modelo denominado DeepFace da empresa Facebook, este modelo acabou sendo reconhecido como VGG-Face, em homenagem aos criadores[15].

Uma das contribuições presentes no artigo original do modelo[37], foi o desenvolvimento de um método de treinamento de modelos que não contasse com uma base extremamente grande. A necessidade deste método de treinamento se deu pela escassez de bases de faces com quantidade suficiente para competir com empresas como Google e Facebook. Desta forma, os autores inovaram e criaram alternativas técnicas para treinar a rede alcançando acurácia similar ou superior a outras redes consideradas estado da arte[15].

Os autores propuseram um método de coleta de dados de face utilizando fontes de conhecimento disponíveis na internet. Ao final, foram coletadas mais de dois milhões de imagens, as quais foram disponibilizadas gratuitamente para a comunidade. Ao comparar com outros modelos, a rede VGG-Face foi treinada com 2,6 milhões de imagens, enquanto a FaceNet do Google foi treinada com 200 milhões de imagens[15].

Como mencionado, os autores inovaram e realizaram o treinamento do reconhecimento facial em duas etapas. A primeira etapa consistia no treinamento da rede com base na base de dados que foi criada, onde na última camada, para classificar o rosto das pessoas, havia uma função SOFTMAX, a qual era tipicamente utilizada em redes de classificação de imagens. Após o treinamento, esta última camada é retirada, e assim a saída da rede passa a ser uma representação da face, em forma de vetor, denominada *face embedding*. Em seguida, o modelo é retreinado, via ajuste fino (do inglês *fine-tunning*), para que, através do cálculo da distância euclidiana, as faces da mesma pessoa tenham distância menor do que as faces de pessoas diferentes. Conforme exposto no trabalho publicado, para realizar o treinamento foi utilizado a função de erro *triplet-loss*[15].

De maneira geral, a rede é uma CNN no estilo VGGNet, ou seja, caracteriza-se por uma rede neural convolucional com filtros (ou *kernels*) pequenos, de tamanho 3x3, funções de ativação ReLU seguidas por *maxpooling*, e camadas totalmente conectadas (*fully-connected*) ao final[37].

A seguir, na Figura 2.15 a rede VGG-Face pode ser melhor visualizada:

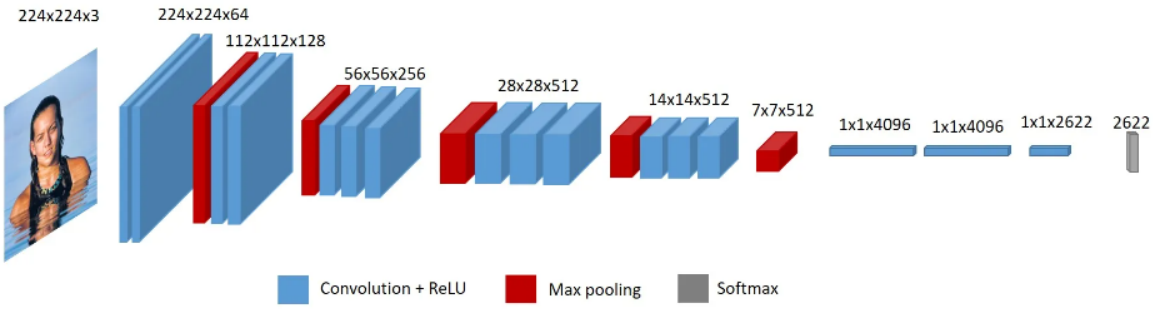


Figura 2.15: Arquitetura VGG-Face (ainda com a camada de classificação ao final). (Imagem extraída de [15])

2.6.6 OpenFace

A modelo OpenFace[39], criado em 2015 por pesquisadores da Universidade Carnegie Mellon, é reconhecido por sua leveza na execução do reconhecimento facial. Um indicador disso é a quantidade de parâmetros do modelo, o qual é de 3,7 milhões de parâmetros treináveis, enquanto o VGG-Face possui 145 milhões e o FaceNet, 22,7 milhões[15].

O modelo foi inicialmente produzido em Lua Torch, sendo posteriormente traduzido para Python com foco em tecnologias móveis, devido a sua velocidade e baixa necessidade de poder computacional. O modelo recebe uma entrada de uma imagem e gera um vetor de representação de 128 dimensões, denominado *face embeddings*. Assim como outros

modelos, as representações podem ser mapeadas no espaço euclidiano e, a partir daí, calcular a similaridade entre as faces[15][39].

2.6.7 ArcFace

Desenvolvido em 2019 por pesquisadores da Universidade Imperial de Londres, o modelo ArcFace deu um passo adiante na acurácia e na estabilidade dos algoritmos de reconhecimento facial[15]. O algoritmo faz uso de uma rede ResNet como backbone e uma função de erro, denominada *Additive Angular Marfin Loss*[17] (ArcFace), a qual aumentou o poder de discriminação e separação das classes (pessoas)[17]. O algoritmo foi disponibilizado publicamente.

A função de erro ArcFace é uma função de margem angular, a qual busca representar o centro da classe no espaço angular, e penalizar os exemplares que se distanciam deste centro. Desta forma, busca-se um aumentar a distância inter-classes (pessoas diferentes) e, consequentemente, diminuir as distâncias intra-classes (variações da mesma pessoa). Ainda, a função busca estabilizar o treinamento da rede quanto utilizada em grandes bases de dados de imagens[17].

A seguir, nas Figuras 2.16 e 2.17 é possível observar o poder discriminativo dos resultados gerados por um modelo treinado com a função de erro ArcFace quando comparada com a função SOFTMAX[17]. Na Figura 2.16 observam-se as margens de decisão das funções de perda, onde a linha pontilhada significa o limite da decisão e a área cinza a margem de decisão[17]. Na Figura 2.17 observa-se oito pessoas (oito centros) sendo representadas no arco. Os pontos indicam os exemplares e a linha o centro da direção de cada identidade. Importante chamar a atenção para o poder de representação da função ArcFace comparada com a SOFTMAX, aproximando as distâncias intra-class e afastando as inter-class[17].

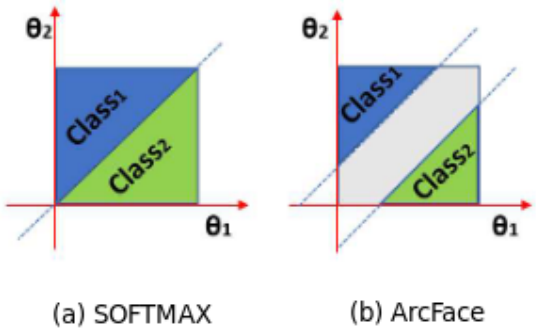
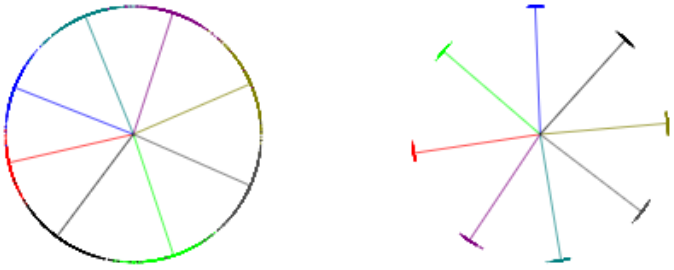


Figura 2.16: Fronteiras de decisão das funções de erro. (Imagem extraída de [17])



(a) SOFTMAX (b) ArcFace

Figura 2.17: Poder de representação da função de erro. (Imagem extraída de [17])

As funções de erro baseadas na função SOFTMAX ou na *triplet-loss* foram muito utilizadas para a tarefa do reconhecimento facial, entretanto, possuem alguns problemas[4, 26, 52]. As funções baseadas em SOFTMAX não são discriminativas o suficiente para tarefas mais complexas de reconhecimento. Já as funções baseadas em *triple-loss* levam a um aumento significativo no número de processamento, quando treinados em bases de dados muito grandes[26]. Assim, as funções baseadas em margem angular surgiram com o propósito de aumentar o poder de discriminação da representação e estabilizar o treinamento[26].

A seguir, na Figura 2.18 verifica-se que a área de interseção entre as curvas normais é mais afastada quando utilizado a função ArcFace em comparação com a *triplet-loss*[17]. Para esta comparação foram selecionados todos os pares positivos e aproximadamente 500 mil pares negativos da base de dados LFW. A área vermelha indica os pares positivos e a área azul os pares negativos[17].

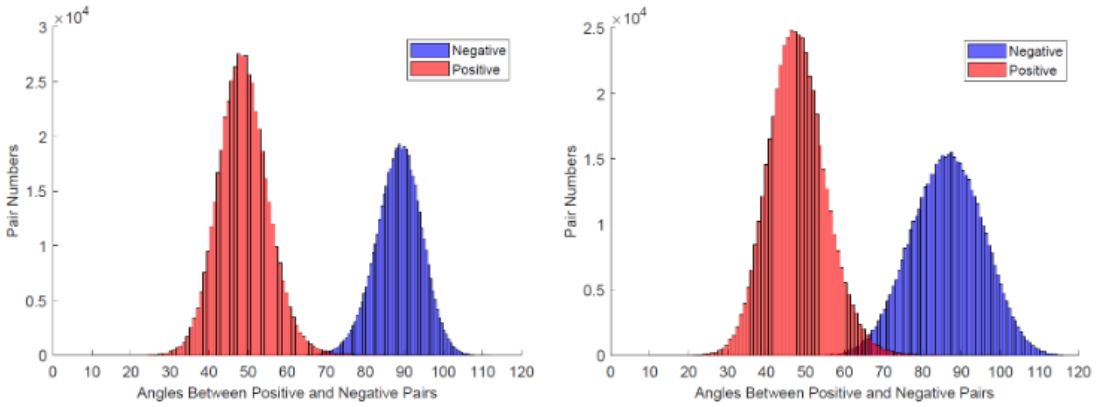


Figura 2.18: Distribuição das funções ArcFace e Triplet-loss. (Imagem extraída de [17])

Conforme mencionado, ArcFace é baseado na rede ResNet-100, a rede recebe uma imagem de entrada no tamanho 112 x 112 *pixels* com 3 (três) canais RGB e gera um vetor unidimensional de saída, com 512 posições, o qual é a representação da face da imagem passada na entrada. Observa-se que a saída de 512 posições é maior em comparação com outros algoritmos considerados estados da arte[17].

Por fim, conforme os resultados expostos no artigo, o algoritmo atingiu uma acurácia de 99,83% quando testado com a base de dados LFW[35], superando os demais algoritmos utilizados na comparação[17].

2.6.8 SFace

Recentemente algumas bases de dados relevantes que eram públicas se tornaram privadas, devido a preocupações éticas e privacidade, tais como VGGFace2[64], MS-Celeb-1M[66], MegaFace[67] e DukeMTMC[68][18]. Neste contexto, surgiu a motivação para o desenvolvimento do algoritmo SFace[18], o qual foi desenvolvido em 2022, pelo instituto Fraunhofer da Alemanha, e usa um *dataset* de faces gerado sinteticamente para treinar modelos de reconhecimento facial.

Para fazer a geração de faces humanas de forma sintética, SFace utiliza uma rede condicional GAN[69], denominada StyleGAN2-ADA[70], a qual foi treinada utilizando dados autênticos das bases de dados públicas. Este modelo é considerado um dos primeiros a utilizar dados sintéticos para a geração de bases de dados para treinamento de modelos de reconhecimento facial[18].

A fim de fazer o treinamento do modelo de reconhecimento facial, foram utilizadas três estratégias diferentes: a primeira era treinar o modelo a partir das faces sintéticas para um problema de classificação de multi-classes, utilizando uma função de perda CosFace[55]; a segunda abordagem era treinar o modelo de reconhecimento facial com as faces sintéticas e fazendo transferência de conhecimento (do inglês *knowledge transfer*) com um modelo pré-treinado em faces autênticas; e a terceira abordagem seria a combinação das duas primeiras[18].

Durante os experimentos, o modelo de rede utilizado como backbone foi o ResNet-50[7], e, ao final, a rede alcançou um escore máximo de 99,13% de acurácia na base de dados LFW[35], se mostrando competitivo e compatível com outros modelos considerados estado da arte[18]. A Figura 2.19 a seguir ilustra a geração sintética das imagens, e as três abordagens de treinamento da rede SFace.

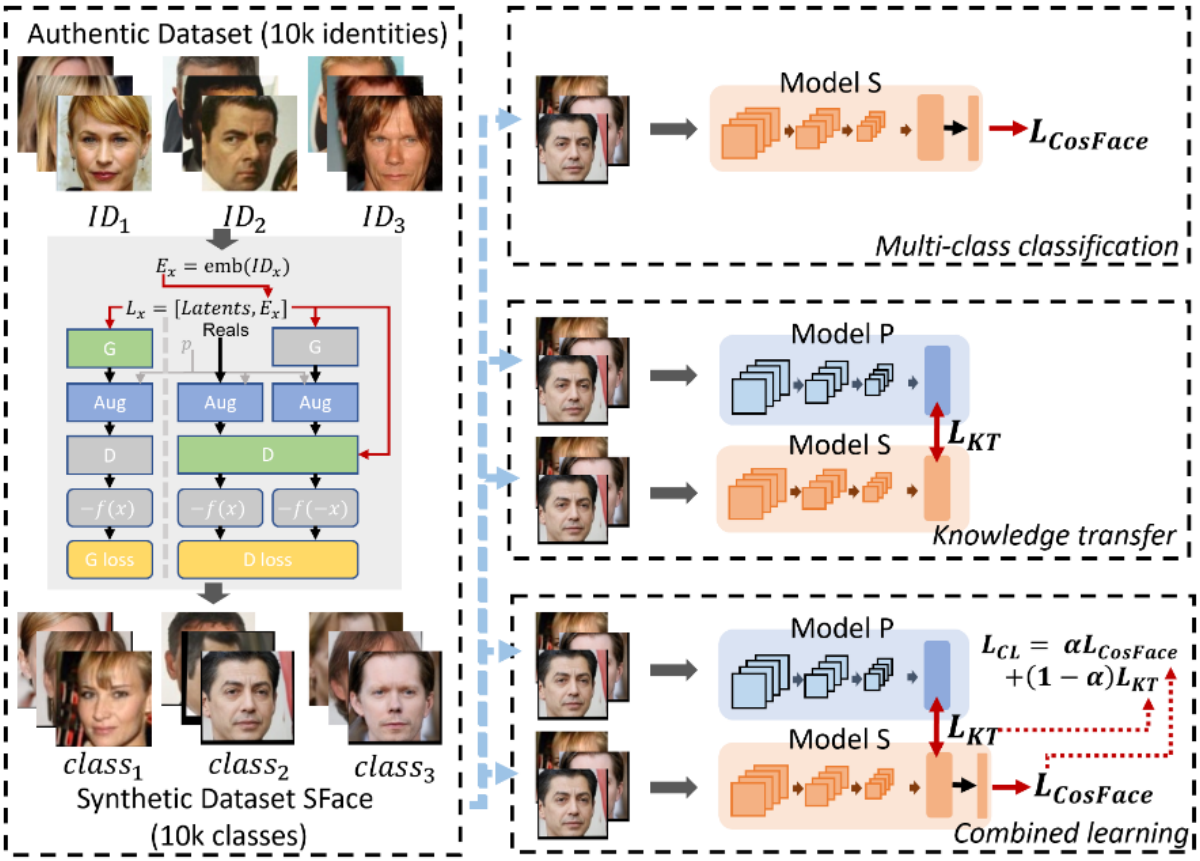


Figura 2.19: Visão geral das abordagens de treinamento da SFace. (Imagem extraída de [18])

2.7 Métricas de Desempenho

Nesta seção, descrevemos as métricas de desempenho utilizadas para medir o impacto da degradação nos algoritmos.

Acurácia: A acurácia é a capacidade do modelo de determinar se o par de imagens é da mesma pessoa ou não. Para calcular esta métrica, deve-se calcular a proporção de positivos verdadeiros e negativos verdadeiros em todos os casos calculados.

A fórmula da métrica Acurácia é dada pela fórmula 4.1, a seguir:

$$\text{Acurácia} = \frac{\text{TN} + \text{TP}}{\text{TN} + \text{FP} + \text{TP} + \text{FN}} \tag{2.2}$$

Precisão: A métrica precisão (do inglês *precision*) é a capacidade do modelo de não classificar um par negativo como sendo um par positivo. Ou seja, a capacidade de evitar predizer que duas imagens são da mesma pessoa, quando, na verdade, não são.

A fórmula da métrica Precisão é dada pela fórmula 2.3, a seguir:

$$\text{Precisão} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2.3}$$

Revocação: A Revocação (do inglês *recall*) é a capacidade do modelo classificar os pares positivos como efetivamente sendo a mesma pessoa.

A fórmula da métrica Revocação é dada pela fórmula 2.4, a seguir:

$$\text{Revocação} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2.4}$$

Para melhor entendimento das métricas, segue abaixo a descrição dos termos utilizados:

Verdadeiro Positivo (do inglês *True Positive* - TP): número de pares corretamente identificados como ”mesma pessoa”;

Falso Positivo (do inglês *False Positive* - FP): número de pares incorretamente identificados como ”mesma pessoa”;

Verdadeiro Negativo (do inglês *True Negative* - TN): número de pares corretamente identificados como ”pessoas diferentes”;

Falso Negativo (do inglês *False Negative* - TP): número de pares incorretamente identificados como ”pessoas diferentes”;

2.8 Funções de Erro

A função de erro é utilizada ao final de uma rede neural para julgar a qualidade da saída da rede[71]. Desta forma, a função tem um papel primordial durante a fase de treinamento, a fim de ajudar a rede neural a encontrar os melhores parâmetros para realizar a tarefa desejada. Conforme explicado anteriormente, para o problema específico de reconhecimento facial, busca-se aproximar a distância entre as imagens de faces de mesma pessoa (*intra-class*) e afastar as imagens de faces de pessoas diferentes (*inter-class*), deixando o resultado da rede mais robusto.[4] Assim, um dos principais trabalhos realizados neste campo é o desenvolvimento de funções de perda adequadas, projetadas especificamente para a tarefa de reconhecimento facial.

Usualmente utilizada em problemas de classificação de imagens, a função de erro denominada SOFTMAX foi a escolha natural para os algoritmos de reconhecimento facial[4]. Entretanto, após seu uso, notou-se que a função não possuía a eficiência necessária para este determinado tipo de aplicação. Assim, após diversas pesquisas, as funções evoluíram e atualmente existem outras mais eficientes, como as baseadas na distância euclidiana, baseadas na margem angular, e até mesmo as variações da própria SOFTMAX[4].

Até 2017 as funções baseadas na distância euclidiana foram muito utilizadas, e com a evolução das pesquisas, as funções baseadas em margem angular/cosseno se tornaram bastante populares[71]. Vale ressaltar que apesar de todas terem o mesmo propósito, a evolução das funções de erro, em regra, busca facilitar o treinamento, bem como permitir uma maior separação entre os grupos de resultados (positivos e/ou negativos)[4][71].

2.8.1 Softmax e variações

Existem diversas variações da função SOFTMAX para o problema de reconhecimento facial, desta forma, citaremos apenas as mais relevantes.

A função denominada *Cross-Entropy Loss* ou SOFTMAX é muito utilizada em aplicações de aprendizagem profunda, e tem se mostrado bastante eficaz na eliminação de valores discrepantes quando utilizadas em tarefas de reconhecimento facial. Uma variação criada, denominada Angular-Softmax Loss[72] realizou muitas modificações na função SOFTMAX a fim de introduzir o aprendizado baseado em margem. Esta função sugeriu um erro baseado em margem angular, denominado Angular-Softmax ou A-Softmax, a qual permite que as CNNs aprendam características angularmente discriminantes. Outra função variante é a Additive-Margin Softmax Loss[73], que, motivado pelo bom desempenho da SphereFace[72] usando A-Softmax, realiza uma margem aditiva para a perda SOFTMAX.

A função SOFTMAX é dada pela Equação 2.5, a seguir:

$$\mathrm{L} = -\sum_{i=1}^{m} \ln \frac{\exp\{W_{y_i}^T \mathbf{x_i} + b_{y_i}\}}{\sum_{i=j}^{n} \exp\{W_j^T \mathbf{x_i} + b_j\}} \tag{2.5}$$

onde Xi é o vetor de características da i$^{\text{ésima}}$ imagem. $\mathrm{W_j}$ é a j$^{\text{ésima}}$ coluna dos pesos e $\mathrm{b_j}$ é o *bias*. O número de classes e o número de imagens é n e m, respectivamente. $\mathrm{y_i}$ é a classe da i$^{\text{ésima}}$ imagem.

A função A-SOFTMAX é dada pela Equação 2.6, a seguir:

$$\mathcal{L}_{\text{ang}} = -\frac{1}{N} \sum_{i} \ln \frac{\exp\left\{||\,\mathbf{x_i}|\,|\cos(m \cdot \theta_{y_i,i})\right\}}{\exp\left\{||\mathbf{x_i}||\cos(m \cdot \theta_{y_i,i})\right\} + \sum_{j \neq y_i} \exp\left\{||\mathbf{x_i}||\cos\left(\theta_{j,i}\right)\right\}} \tag{2.6}$$

onde N é o total de exemplares de treinamento, x e y são o vetor de entrada e a classe para o $n^{ésima}$ exemplar de treinamento. $\theta_{y_i,i}$ é o angulo entre x_i e o vetor de pesos, e deve ser um valor entre $[0, \pi/m]$.

2.8.2 Distância Euclidiana

Trata-se de uma função que trabalha as imagens em um espaço euclidiano e busca reduzir as intra-variações (variações da mesma pessoa) ao passo que aumenta as inter-variações (varições de pessoas diferentes), sendo considerado o estado da arte até 2017[4]. As principais funções deste grupo são as denominadas *contrastive loss*[71] e *triplet loss*[38].

A *contrastive loss* realiza o treinamento com pares de imagens (pares positivos de mesma classe ou pares negativos de classes diferentes). Já a função *triplet loss*, introduzida pela rede FaceNet[38] (criada pela empresa Google), requer três imagens para o treinamento, sendo uma âncora, a qual é comparada com um exemplo de mesma classe (positivo) e de outra classe (negativo)[4].

As distâncias entre a âncora e o positivo são minimizadas enquanto entre a âncora e o negativo são maximizadas. Durante o treinamento, essas três imagens (âncora, positiva e negativa) são fornecidas para o modelo como se fossem uma única imagem (as três imagens são concatenadas). Conforme mencionado, a distância entre a âncora e o exemplo positivo deve ser menor do que entre a âncora e o negativo.

A diferença principal entre a *contrastive loss* e a *triplet loss* reside no fato de a primeira considerar a distância absoluta entre os pares de imagens fornecidas durante o treinamento, enquanto a segunda considera a distância relativa entre elas[71][4][3].

A função *Contrastive Loss* é dada pela Equação 2.7, a seguir:

$$L = \frac{1}{2N} \sum_{i}^{N} y_i \cdot d_i^2 + (1 - y_i) \cdot max(m - d_i, 0)^2 \tag{2.7}$$

onde y_i é uma variável binária indicando se o par i é similar (1) ou diferente (0), d_i é a distância entre os *embeddings*, N é o total de pares e m é a margem.

Já a função *Triplet Loss* é dada pela Equação 2.8, a seguir:

$$\mathcal{L} = \frac{1}{N}\sum_{i}^{N} max(||f(A_i) - f(P_i)||^2 - ||f(A_i) - f(N_i)|\,|^2 + m, 0) \tag{2.8}$$

onde A_i é o exemplar âncora, P_i é o exemplar positivo, N_i, o exemplar negativo e m é a margem.

Apesar da evolução no treinamento, a *contrastive loss* e a *triplet loss* apresentaram instabilidade durante o treinamento, e novas pesquisas na área propuseram outras funções nesta seara, tais como a *center loss*, que trabalha com um centro da classe e penaliza as imagens distantes deste centro. Porém, a desvantagem desta função é o consumo massivo de memória de GPU[3].

A Figura 2.20 ilustra a forma de treinamento da *triplet loss*, onde busca-se aproximar a imagem de mesma classe da âncora e afastar a imagem de classe diferente.

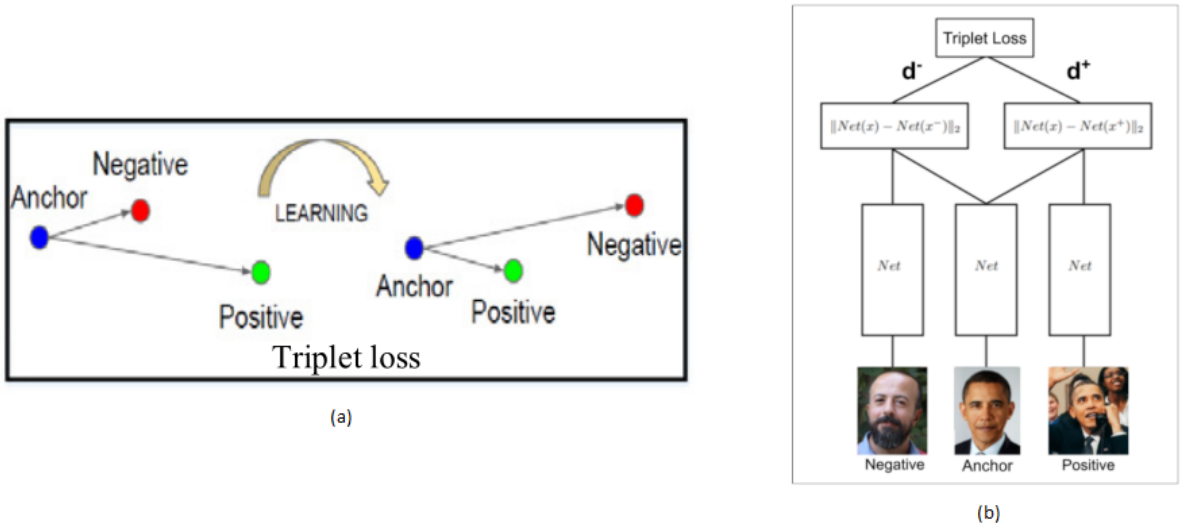


Figura 2.20: Triplet Loss: (a) Treinamento, (b) Âncora/Positivo/Negativo. (Imagens extraídas de [19][20])

2.8.3 Margem Angular

A fim de separar os exemplares de forma mais contundente e evitar classificações errôneas, a função de erro denominada *Sphere Face*[72] foi proposta. A *Sphere Face* utilizou uma função angular SOFTMAX (A-SOFTMAX) para treinar CNNs realizando o aprendizado das características angularmente discriminantes[3].

Em seguida, uma nova função denominada CosFace[55] foi criada. Com o mesmo objetivo das demais funções, o qual era minimizar as distâncias intra-classe (distância entre mesma pessoa) e maximizar as inter-classes (distâncias entre pessoas diferentes), a CosFace buscou uma perspectiva diferente. A função de erro SOFTMAX foi reformulada, sendo introduzida a margem de cosseno a fim de maximizar ainda mais a margem de decisão no espaço angular, Figura 2.21. Em resumo, com a CosFace foi possível separar as características aprendidas com uma distância angular/cosseno maior, permitindo assim, uma acurácia maior ao analisar a distância angular/cosseno[3][71].

A função CosFace é dada pela Equação 2.9, a seguir:

$$\mathcal{L}_{\text{lmc}} = -\frac{1}{N}\sum_{i=1}^{N}\ln\frac{\exp\{s\cdot(\cos(\theta_{y,i})-m)\}}{\exp\{s\cdot(\cos(\theta_{y,i})-m)\}+\sum_{j\neq y_i}\exp\{s\cdot(\cos(\theta_{j,i}))\}} \tag{2.9}$$

onde N é o número de exemplares, θ_i é o ângulo entre o vetor de entrada e o peso associado a classe y_i, m é a margem e s é o fator de escala.

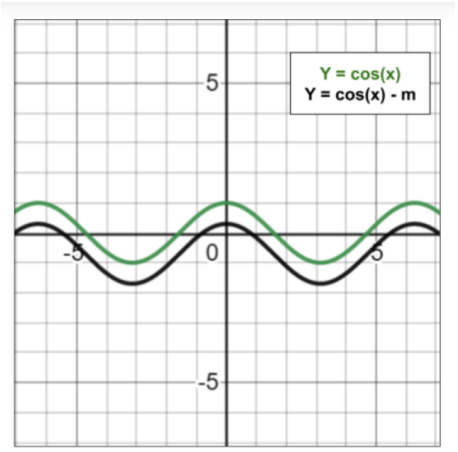


Figura 2.21: Margem adicionada pela função CosFace. (Imagem extraída de [20])

Para vencer as dificuldades de cálculo da margem angular, a função denominada ArcFace[17] foi desenvolvida. A ArcFace possui base em outras funções de perda, e sugeriu uma nova margem para o angulo cosseno, que os autores afirmam ser mais rigorosa para classificação[3]. Segundo os experimentos dos autores[17], a margem angular da ArcFace representa uma melhor interpretação geométrica em comparação com SphereFace e CosineFace, Figura 2.22. Assim, a ArcFace tornou o processo de treinamento mais estável e aumentou o poder de discriminação dos modelos de reconhecimento facial. Atualmente é uma função considerada estado da arte, amplamente utilizada em diversos modelos disponíveis na área[3][71][26]. A função ArcFace é dada pela Equação 2.10, a

seguir:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N} \ln \frac{\exp\{s \cdot \cos(\theta_{y,i} + m)\}}{\exp\{s \cdot \cos(\theta_{y,i} + m)\} + \sum_{j \neq y_i} \exp\{s \cdot (\cos(\theta_{j,i}))\}} \tag{2.10}$$

onde N é o número de exemplares, θ_i é o ângulo entre o vetor de entrada e o peso associado a classe y_i, m é a margem e s é o fator de escala.

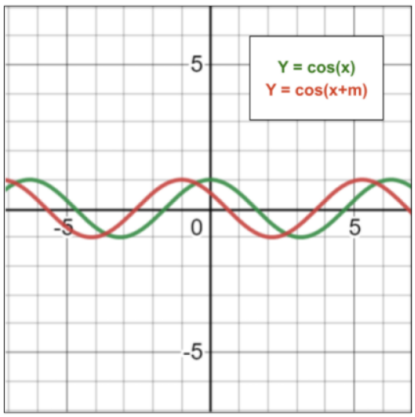


Figura 2.22: Margem adicionada pela função ArcFace. (Imagem extraída de [20])

A Figura 2.23 demonstra a margem estabelecida entre as classes[17].

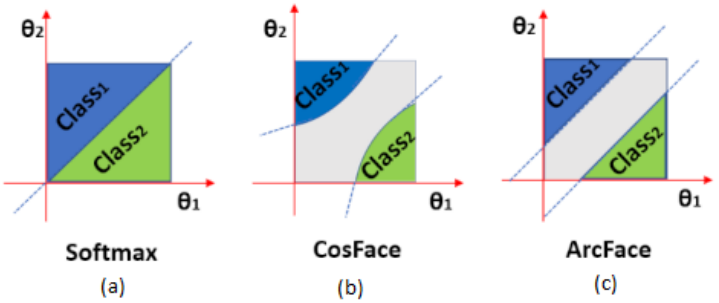


Figura 2.23: Comparação de margens: (a) Softmax, (b) CosFace e (c) ArcFace. (Imagem extraída de [17])

2.9 Qualidade da Imagem

Com o avanço da tecnologia, viu-se um avanço no uso de imagens digitais como meio de representação e comunicação de informações. Observou-se também uma evolução nas pesquisas e na literatura dedicadas a métodos para melhorar a nitidez das imagens. Entretanto, a qualidade das imagens digitais é raramente perfeita, já que estão sujeitas a diversas distorções durante a aquisição, compressão, transmissão, processamento e reprodução[74][2].

Um sistema que possa identificar e quantificar as degradações da qualidade da imagem é fundamental para manter, controlar e melhorar a qualidade das imagens durante os processos de aquisição, gerenciamento, comunicação e processamento. Assim, o desenvolvimento de sistemas automáticos eficazes na avaliação da qualidade da imagem é uma necessidade para atender o problema supramencionado[2].

Entretanto, apesar do tópico avaliação de qualidade da imagem não ser novo, estando presente nas pesquisas acadêmicas já há algumas décadas, observa-se um número relativamente menor de publicações na área. Assim, há de se ressaltar que o campo de avaliação da qualidade da imagem ainda se encontra em estado inicial, necessitando de muito desenvolvimento e pesquisa para atingir um nível de maturidade semelhante a outras áreas do mesmo segmento[2][21].

O conceito de qualidade é subjetivo, por décadas vem sendo discutido, e apesar de alguns trabalhos buscarem objetivar o assunto, segue sem uma definição pacificada. As ISOs 29794-5[75] e 39794-5[53] segmentam o conceito de qualidade em três aspectos: Caráter, fidelidade e utilidade. Em relação à biometria facial[2][4], estes conceitos podem ser descritos da seguinte forma:

Caráter: atributos inerentes à característica biométrica de origem que está sendo adquirida, que não podem ser controlada durante o processo de aquisição;

Fidelidade: fidelidade reflete o grau de similaridade com a fonte biométrica; e

Utilidade: significa o quão boa é a imagem para um sistema de biometria. É influenciada pelo caráter e fidelidade. Pode ser utilizado como o escore que o algoritmo de qualidade atribui à imagem.

Na Figura 2.24 pode-se observar diversos níveis de qualidade da imagem da face.

Figura 2.24: Imagens de faces com qualidades diferentes. (Imagem extraída de [21])

2.9.1 Algoritmos de Avaliação da Qualidade da Imagem da Face

A avaliação de qualidade da face se refere ao processo de receber uma imagem como entrada para um algoritmo e, retornar um escore de qualidade (do inglês *quality score* - QS) como saída[4][76], conforme ilustrado na Figura 2.25. Desta forma, um algoritmo de avaliação de qualidade da face (do inglês *Face Image Quality Assessment Algorithm* - FIQAA)[2] é a automatização deste processo.

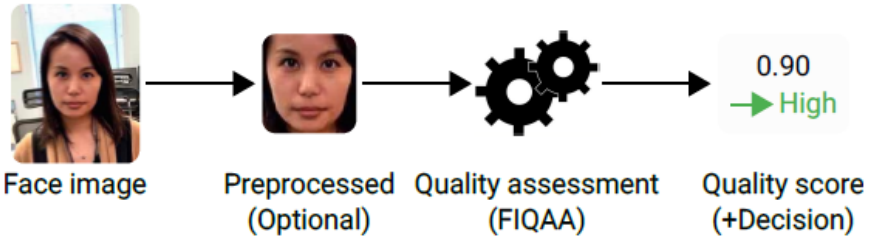


Figura 2.25: Sistema de avaliação de qualidade da face. (Imagem extraída de [2])

A avaliação automática da qualidade do exemplar possui diversas utilidades, pois pode detectar imagens de faces de baixa qualidade e tomar decisões conforme o contexto da situação. O algoritmo pode ser empegado em aplicações multi-quadros, onde há a necessidade de se escolher a melhor imagem da face, dentre várias disponíveis; aplicações onde o usuário precisa de um retorno a respeito da própria captura, como em aplicações de autenticação; aplicações onde a imagem precisa de um limiar mínimo de qualidade para poder ser considerada útil, rejeitando aquelas que não atingem tal escore, entre outras[2][21][76]. Desta forma, a automatização da avaliação da qualidade da face de uma imagem pode ser o diferencial entre o sucesso ou a inviabilização de uma aplicação.

2.9.2 Escores de Qualidade

Em relação à qualidade da imagem, há de se distinguir entre qualidade da imagem na concepção mais genérica e qualidade da imagem da face. Apesar de ser um campo de estudo pesquisado há mais tempo, a Avaliação da Qualidade da Imagem (do inglês *Image Quality Assessment* - IQA)[74] apresenta distinções importantes quando aplicada às faces, de forma que esta área (IQA) e FIQA (do inglês *Face Image Quality Assessment* - FIQA), embora sobrepostas, não devem ter sua área de aplicação confundida[2].

Ainda, normalmente os escores de qualidade[2] de um algoritmo de avaliação da qualidade da imagem (*Image Quality Assessment Algotithm* - IQAA) tem por objetivo indicar o nível de qualidade da imagem para a percepção subjetiva humana, ou seja, os valores objetivos gerados por IQAA normalmente tem a intenção de indicar um percentual de

qualidade subjetivo de acordo com os padrões humanos[2]. Já um algoritmo de avaliação de qualidade da imagem da face (*Face Image Quality Assessment Algorithm* - FIQAA) busca indicar um escore de qualidade para os padrões dos algoritmos de reconhecimento facial, ou seja, o escore é objetivo em relação à utilidade daquela imagem quando fornecida como entrada para um sistema de reconhecimento facial[2]. Um FIQA, em geral, retorna como avaliação de uma imagem um valor escalar único, ou um vetor de valores de qualidade medindo diferentes elementos relacionados àquela imagem[76].

Assim, é possível que um FIQAA possa ser mais acurado para um determinado sistema de reconhecimento facial (*Face Recognition* - FR) específico[21] do que para outro. Entretanto, apesar destes casos, é esperado, em termos gerais, que um FIQAA funcione para diversos algoritmos de FR[2].

2.10 Degradações em Imagens Digitais

A degradação de uma imagem digital ocorre quando a imagem perde sua informação, resultando em uma perda de qualidade. Isto pode ocorrer devido a diversos fatores como defeitos no sensor de captura, interferências durante a transmissão da imagem, aplicação de métodos de compressão, inserção de ruídos, entre outros[77].

Algumas degradações podem ser restauradas durante o processo de restauração da imagem. Este processo tem por objetivo restaurar imagens corrompidas, considerando um conhecimento anterior relativo ao processo que as degradou[77]. Entretanto, não faz parte do escopo deste trabalho tratar do fenômeno da restauração, assim, vamos focar no processo e nas formas de degradação da imagem.

A fim de avaliar o impacto da degradação no fluxo de um reconhecimento facial, foram utilizadas imagens degradadas de bases de dados públicas. Nos subitens seguintes, descrevemos as degradações utilizadas no experimento.

Parte das degradações aqui estudadas, também foram exploradas nos trabalhos [78, 31, 79, 80, 81, 82].

Na Figura 2.26 pode-se observar as degradações puras e na Figura 2.27, as degradações em sequência.

2.10.1 Ruído

O ruído é uma variação aleatória das informações de brilho ou cor presentes em uma imagem. Normalmente, o ruído é gerado no processo de captura da imagem e é originário dos sensores e componentes eletrônicos do sistema de aquisição. Alguns tipos de ruído são: ruído gaussiano; ruído impulsivo ("sal e pimenta"); ruído uniforme; ruído periódico; ruído quântico; e ruído *speckle*[31][78].

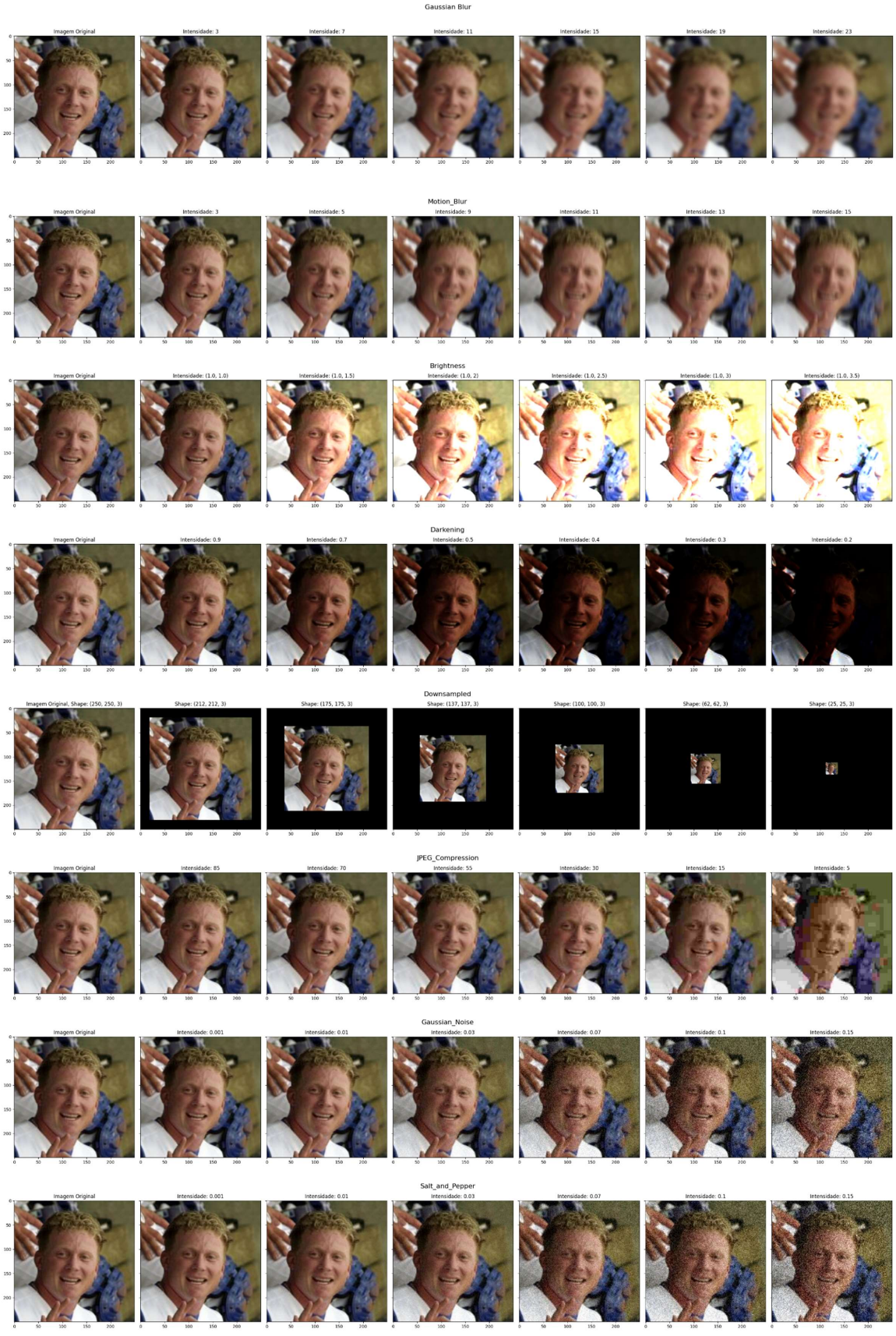


Figura 2.26: Degradações puras. (Imagens geradas a partir da base de dados LFW)

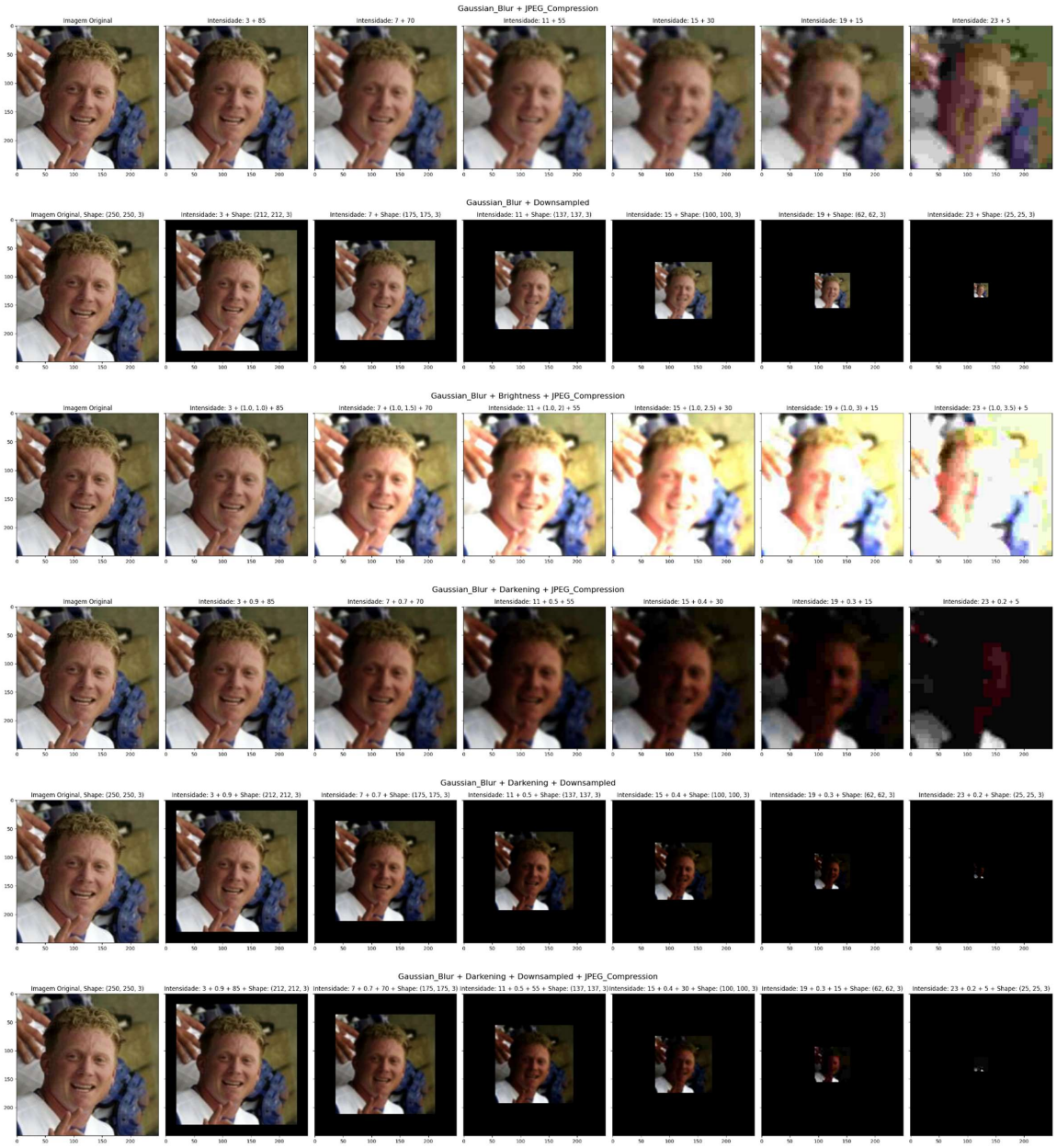


Figura 2.27: Degradações em sequência. (Imagens geradas a partir da base de dados LFW)

Ruído gaussiano: O ruído gaussiano pode ser adicionado em uma imagem durante sua captura, devido a algum problema no sensor, ou ainda durante a transmissão por algum canal[31][78][80][79][79];

Ruído sal e pimenta: Outro ruído muito comum é o denominado sal e pimenta. Este é um ruído de impulso tipicamente observável em imagens devido a distúrbios intensos. Ele se caracteriza por substituir valores de *pixels* originais por *pixels* brancos e pretos, de forma aleatória[31][78][80][79].

2.10.2 Borramento

Borramento de uma imagem significa uma imagem suavizada, ou seja, as transições abruptas são atenuadas. Este tipo de processamento digital pode ser feito para minimizar ruídos, entretanto apresenta o efeito de borramento da imagem[31]. O borramento é uma degradação muito comum de ser encontrada, principalmente em imagens capturadas em ambiente não controlado. Como consequência, a imagem pode perder nitidez, devido à suavização das altas frequências. Alguns exemplos são os filtros de média e filtros gaussianos[78].

Borramento gaussiano: A geração de imagens borradas (borramento gaussiano) se deu com o uso de filtros gaussianos com diferentes tamanhos de janelas de filtro (*kernel*)[31][78][80][79];

Borramento de movimento: O borramento de movimento ocorre, geralmente, devido à falta de estabilidade da câmera, ou até devido ao movimento do objeto/pessoa que está sendo filmado[80][52]. Este tipo de borramento é bastante observado quando a filmagem é realizada a partir de dispositivos móveis, devido à falta de estabilidade da pessoa que realiza as filmagens. O estudo do borramento de movimento também foi observado na pesquisa de [81].

2.10.3 Luminosidade

A luminosidade pode ser inserida de diversas formas em uma captura. Usualmente, em locais públicos pode se dar pelo brilho, durante incidência de luz solar, ou pelo escurecimento, devido à falta de luz durante a noite.

Brilho e Escurecimento: Para avaliar o comportamento dos sistemas com diversos níveis de exposição, foram realizadas mudanças gradativas de luminosidade nas imagens, tanto de brilho quanto de escurecimento[31].

2.10.4 Tamanho da Imagem

Redução de Tamanho: Em capturas realizadas em ambiente não controlado, é comum o suspeito encontrar-se longe do ponto de captura da câmera, sendo a imagem da face representada por poucos *pixels*. Para este experimento, foram realizadas reduções gradativas de tamanho nas imagens das bases de dados[79].

2.10.5 Compressão

A compressão da imagem é um processo aplicado nos arquivos de imagem visando reduzir o tamanho de armazenamento, sem haver perda da informação da imagem até um determinado limiar de compressão. Ao reduzir o tamanho dos arquivos, a compressão permite armazenar mais arquivos no disco rígido, menor consumo de memória ao carregar o arquivo, transmissão mais rápida pela banda, menor congestionamento na rede de dados, entre outros. Por isso, a compressão é um processo que está presente na maioria dos casos onde há necessidade de transmissão e/ou armazenamento de arquivos de imagens[77].

Os métodos de compressão se subdividem em duas categorias, sendo: com perda de informação e sem perda de informação. A compressão com perda de informação tem maior poder de reduzir a quantidade de bytes de uma imagem, já que remove, de forma permanente, as informações de menor valor em uma imagem. O ponto negativo é que a compressão com perdas pode reduzir a qualidade da imagem, apresentando inclusive imagens finais com distorção ou com "defeitos", denominados artefatos de compressão. Entretanto, mesmo com a compressão com perdas, quando aplicada de forma cuidadosa e respeitando as características da imagem, a qualidade da imagem pode ser mantida[77][80].

Cabe ressaltar que o processo de compressão com perdas é irreversível, desta forma, as informações perdidas durante o processo não podem ser recuperadas posteriormente. Isto se torna mais evidente quando a mesma imagem sofre diversas compressões, por diversos métodos diferentes, muitas vezes tornando o resultado final inteligível[77].

A compressão com perdas é reconhecidamente necessária no cenário da internet, onde se não houvesse compressão das imagens, algumas aplicações seriam praticamente inviabilizadas devido ao alto tempo para transmissão destes dados. Um exemplo muito comum de compressão com perdas é o padrão JPEG, o qual é muito utilizado em aplicações web e em fotografia digital[80][77].

A outra categoria de compressão é a compressão sem perdas. Nesta categoria estão os algoritmos que realizam a compressão, mas não removem informações, consequentemente não reduzem a qualidade da imagem. Estes algoritmos podem reduzir o tamanho do arquivo gerando códigos que podem ser retornados para o valor da informação original, garantindo assim que não haja degradação ou distorção na imagem[77].

Em regra, este tipo de compressão é utilizado onde a qualidade da imagem é mais importante do que a capacidade de armazenamento do arquivo ou a velocidade de transmissão na rede. A desvantagem é que a capacidade de redução do arquivo é muito inferior em comparação aos algoritmos que realizam a compressão com perdas. Entre os algoritmos que utilizam este método de compressão, o formato PNG é um dos mais conhecidos[77].

Compressão JPEG: A fim de analisar o impacto da compressão nos modelos de reconhecimento facial, foi utilizado o algoritmo JPEG, o qual é um algoritmo de compressão com perdas. Assim, é possível indicar ao algoritmo o nível de compressão desejado para ser aplicado na imagem, onde, quanto maior o coeficiente aplicado, maior é a compressão, e consequentemente, maior a degradação da imagem resultante[77][80][82].

2.10.6 Degradações sequenciais

Para simular ambientes reais, onde normalmente há mais de uma degradação incidindo sobre a imagem, foram realizadas degradações em sequência nas imagens antes de serem fornecidas aos sistemas de reconhecimento facial. Desta forma, a imagem original foi degradada com um tipo de degradação, e em seguida, a imagem resultante foi submetida a novo tipo de degradação[79][81].

Conforme mencionado, este ponto do trabalho consiste em uma tentativa de aproximação ao cenário real, onde, por exemplo, o suspeito está longe do ponto de captura da câmera, a noite, incidindo pouca iluminação, o equipamento de captura possui baixa resolução e o arquivo ainda é comprimido pelo equipamento. Assim, explorando degradações em sequência, podemos observar o comportamento dos algoritmos ao longo do experimento.

Degradações utilizadas:

Borramento Gaussiano → Compressão JPEG: Primeiro foi realizado a degradação da imagem com o borramento gaussiano, em sequência, a imagem degradada foi submetida à compressão JPEG. O objetivo da escolha desta sequência de degradação é a simulação de imagens produzidas por câmeras de baixa qualidade (com baixa nitidez) e em seguida sofreram um processo de compressão para serem transmitidas;

Borramento Gaussiano → Redução de Tamanho: Inicialmente, foi realizado a degradação da imagem com o borramento gaussiano, em sequência, a imagem degradada foi reduzida de tamanho. O objetivo da escolha desta sequência de degradação é a simulação de imagens produzidas por câmeras de baixa qualidade (com baixa nitidez) e cujo alvo de interesse esteja longe do ponto de captura. Esta sequência

de degradações foi objeto de estudo de [79], entretanto, no estudo citado, primeiro foi aplicado a redução de tamanho e posteriormente o borramento gaussiano;

Borramento Gaussiano → Brilho → Compressão JPEG: A imagem foi degradada com o uso de borramento gaussiano, e, em sequência, a imagem resultante foi exposta a uma intensidade de brilho e, finalmente, comprimida com algoritmo JPEG. O objetivo da escolha desta sequência de degradação é a simulação de imagens produzidas por câmeras de baixa qualidade (com baixa nitidez) e cujo alvo de interesse esteja exposto à alta luminosidade, como luz solar, por exemplo, e em seguida comprimida para transmissão;

Borramento Gaussiano → Escurecimento → Compressão JPEG: Nesta sequência, a imagem foi degradada com o uso de borramento gaussiano, e, em sequência, a imagem resultante foi exposta a um processo de escurecimento e, finalmente, comprimida com algoritmo JPEG. O objetivo da escolha desta sequência de degradação é a simulação de imagens produzidas por câmeras de baixa qualidade (com baixa nitidez) e cujo alvo de interesse esteja exposto à baixa luminosidade, como capturas noturnas, por exemplo, e em seguida comprimida para transmissão;

Borramento Gaussiano → Escurecimento → Redução de tamanho: Nesta sequência, a imagem foi degradada com o uso de borramento gaussiano, e, em sequência, a imagem resultante foi exposta a um processo de escurecimento e, finalmente, reduzida de tamanho. O objetivo da escolha desta sequência de degradação é a simulação de imagens produzidas por câmeras de baixa qualidade (com baixa nitidez) e cujo alvo de interesse esteja exposto à baixa luminosidade, como capturas noturnas, por exemplo, e longe do ponto de captura (câmera);

Borramento Gaussiano → Escurecimento → Redução de tamanho → Compressão JPEG: Nesta sequência, a imagem foi degradada com o uso de borramento gaussiano, e, em sequência, a imagem resultante foi exposta a um processo de escurecimento, depois a uma redução de tamanho e, finalmente, comprimida com algoritmo JPEG. O objetivo da escolha desta sequência de degradação é a simulação de imagens produzidas por câmeras de baixa qualidade (com baixa nitidez) e cujo alvo de interesse esteja exposto à baixa luminosidade e longe do ponto de captura, como suspeito distante e capturas noturnas, por exemplo, e em seguida comprimida para transmissão;

2.11 Bases de Imagens Faciais

Para avaliar a influência das degradações, foram usados conjuntos de imagens conhecidos e amplamente utilizados como forma de benchmark entre algoritmos, a fim de estabelecer um padrão de comparação mas tangível:

2.11.1 LFW

O conjunto de dados LFW (Labeled Faces in the Wild)[35], é um conjunto de dados de referência *benchmark* conhecido e usado no campo da visão computacional e do reconhecimento facial. Foi criado em 2007. Contém uma coleção de imagens de rostos coletadas da internet, o que o torna desafiador para tarefas de reconhecimento, devido à sua ampla variabilidade em termos de iluminação, pose, expressão, idade, oclusão e etnia.

O conjunto de dados LFW possui 13.233 imagens rotuladas de rostos, pertencentes a 5.749 indivíduos diferentes (classes). Cada indivíduo é representado por um número variável de imagens, variando de uma até mais de 100 imagens. As imagens exibem significativas variações de iluminação, expressões faciais, orientação da cabeça, oclusões, acessórios, elementos de fundo, entre outras. Essa diversidade torna o conjunto de dados realista e desafiador para algoritmos de reconhecimento facial.

O *dataset* é organizado de maneira em que se possa identificar imagens que pertencem à mesma pessoa, e imagens que pertencem a pessoas diferentes, permitindo assim os estudos elaborados neste trabalho.

2.11.2 FEI

O conjunto de dados FEI (Facial Expression Images)[83], é uma coleção de imagens focada no estudo e na análise de expressões faciais. Esse conjunto de dados foi desenvolvido para apoiar pesquisas relacionadas ao reconhecimento de emoções e expressões em faces humanas. O *dataset* possui faces obtidas entre junho de 2005 e março de 2006, no Laboratório de Inteligência Artificial, em São Bernardo do Campo, São Paulo, Brasil. Ainda, o dataset conta com 14 imagens para cada um dos 200 indivíduos, gerando um total de 2.800 imagens.

As imagens do FEI possuem alta qualidade e resolução, contribuindo para a acurácia dos algoritmos. Ainda, as imagens são coloridas e foram capturadas contra um fundo homogêneo. A escala pode variar cerca de 10% e o tamanho original de cada imagem é 640x480 pixels. Todos os rostos são representados principalmente por estudantes e funcionários da FEI, entre 19 e 40 anos, com aparências, penteados e adornos distintos.

2.11.3 SCFace

O conjunto de dados SCface (Surveillance Cameras Face Database)[84] é um conjunto de dados de referência amplamente utilizado para pesquisas de reconhecimento facial. Foi criado pelo Laboratório de Processamento de Sinais (LTS5) da École Polytechnique Fédérale de Lausanne (EPFL) na Suíça.

O conjunto de dados SCface contém imagens faciais de 130 sujeitos, com 52 imagens por sujeito, em diferentes câmeras, captura de distância e resoluções, resultando em 6.760 imagens. As imagens mais aplicáveis a este estudo foram relativas à câmera denominada mugshot rotation all, que possui 1.170 imagens.

2.11.4 GUFG

O GUFD (Glasgow Unfamiliar Face Database)[85] é um conjunto de dados de faces de pessoas desconhecidas. Ele foi criado pelo Face Perception Lab da Universidade de Glasgow, no Reino Unido. O GUFD contém 5 fotos de 303 indivíduos, gerando um total de 3.028 imagens, sendo anotado em relação aos indivíduos, ou seja, permite identificar imagens da mesma pessoa e/ou pessoas diferentes. Todas as imagens são coloridas e foram capturadas em ambiente controlado, com iluminação uniforme. As dimensões espaciais das imagens são cerca de 2.288 x 1.712 *pixels*, garantindo assim boa qualidade das fotografias. Os participantes da coleta foram instruídos a olhar diretamente para a câmera e manter uma expressão facial neutra.

2.12 Ambientes controlados e não-controlados

Normalmente, durante uma aplicação de autenticação de pessoas, uma imagem padrão da face da pessoa é capturada de forma antecipada, durante o cadastro da aplicação. Em determinado momento posterior, uma pessoa é apresentada, e cabe ao sistema capturar uma imagem da face dessa pessoa (face questionada) e comparar com as faces das pessoas previamente cadastradas (faces padrão), a fim de verificar se existe tal pessoa no "banco de faces". Este processo já foi descrito e trata-se da Identificação da Face (*face identification*) ou 1-N *face matching*[3][49][4].

Este processo de aquisição da face pode ser dividido em 2 grandes grupos, sendo: aquisição em ambiente controlado ou aquisição em ambiente não controlado. No caso da aquisição em ambiente controlado, a pessoa é cooperativa durante a aquisição, isto significa que a posição da cabeça é ajustada de forma frontal à câmera, é mantida a expressão neutra da face e as condições do ambiente são controladas, tais como luminosidade e

fundo (textura). Este ambiente é tipicamente encontrado em fotos para documentos oficiais, como passaporte, por exemplo[4][2].

A aquisição em ambiente não controlado tende a ser mais complicada, já que a pessoa não está cooperativa. A expressão "não cooperativa", neste caso, se aplica tanto para a pessoa que está indiferente para o processo de captura quanto para a pessoa que intencionalmente não permite que a captura seja realizada. Ainda, nesta classificação, as condições do ambiente não podem ser controladas. Como exemplo, as câmeras de vigilância fixadas em vias públicas[4][2].

Ainda no contexto da aquisição da imagem, existem outros cenários que fogem dos extremos supramencionados, como, por exemplo, uma pessoa com um aparelho celular, buscando fazer o cadastro em algum aplicativo que exija o reconhecimento facial. Neste caso a pessoa está cooperativa, possui a intenção de fornecer a imagem, entretanto o ambiente pode não ser o mais propício para a captura da foto, havendo desbalanceamento de luminosidade ou alguma textura no plano de fundo da imagem[4][2].

2.13 Espinha Dorsal dos Modelos de Aprendizagem Profunda

A espinha dorsal (do inglês *backbone*) de um modelo consiste em várias camadas (entrada, ocultas e saída), onde cada camada contém um conjunto de neurônios que realizam operações matemáticas nos valores de entrada para produzir valores de saída. As conexões entre essas camadas formam os pesos do modelo, que são ajustados durante o treinamento para otimizar o desempenho.

A seguir, descreveremos a espinha dorsal dos modelos de aprendizagem profunda utilizados na tarefa de identificação da degradação presente na imagem.

ResNet

A arquitetura ResNet foi descrita na subseção 2.3.2, desta forma faremos apenas uma descrição concisa neste tópico. A família ResNet (Rede Neural Residual) é uma série de arquiteturas de aprendizagem profunda desenvolvidas para aprimorar o treinamento e o desempenho de redes neurais profundas. A ideia fundamental por trás do ResNets é a introdução de blocos residuais, também conhecidos como *skip connections*, para resolver o problema do desaparecimento do gradiente (do inglês *vanish gradient*) à medida que a rede se aprofunda.

Os blocos residuais introduzem conexões que permitem que a informação original flua diretamente de uma camada para outra, em vez de ser totalmente transformada pela

camada intermediária. Isso ajuda a evitar uma diminuição no desempenho à medida que a rede aumenta em profundidade[7].

A arquitetura ResNet está organizada em diferentes profundidades, como ResNet-18, ResNet-34, ResNet-50, ResNet-101, ResNet-152, entre outras. Os números após "ResNet" indicam o número total de camadas na rede. Quanto maior o número, mais profunda é a rede. Para este experimento, foram utilizados ResNet-18, ResNet-50 e ResNet-152.

DenseNet

A ideia central por trás das DenseNets é promover conexões densas entre as camadas da rede, permitindo que as informações passem diretamente entre todas as camadas de maneira altamente interconectada. A estrutura DenseNets visa abordar alguns desafios associados a redes profundas, como o problema do desaparecimento do gradiente e o *overfitting*. Ou seja, para melhorar o fluxo de informação entre as camadas foram introduzidas conexões diretas de qualquer camada para todas as camadas subsequentes, conforme Figura 2.28.

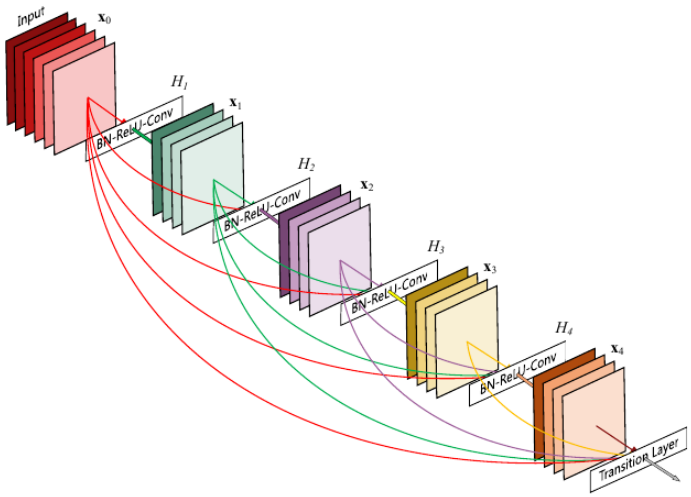


Figura 2.28: Conexões da arquitetura DenseNet.

As DenseNets são amplamente utilizadas para tarefas de visão computacional, incluindo classificação de imagens, detecção de objetos e segmentação semântica. A abordagem de redes neurais densas mostrou melhorias significativas no treinamento eficiente de redes profundas e na utilização eficaz de informações em múltiplas camadas[86].

A família DenseNet possui variantes, como DenseNet-121, DenseNet-169, DenseNet-201, que diferem principalmente em profundidade e número de parâmetros. Para este experimento, foram utilizados DenseNet-121 e DenseNet-201.

VGG

A arquitetura VGG também foi descrita na subseção 2.3.2, e, igualmente, faremos apenas uma descrição concisa neste tópico. A rede neural VGG (Visual Geometry Group) é uma arquitetura de rede neural convolucional (CNN) desenvolvida pelo Visual Geometry Group da Universidade de Oxford, em 2014.

Em relação a outras redes, a VGG-19 é caracterizada por sua simplicidade e arquitetura profunda. Ainda, ela demonstrou que aumentar a profundidade da rede poderia levar a um melhor desempenho em tarefas de reconhecimento de imagem[10]. VGG-19, que é usado neste experimento, é uma versão com mais camadas de profundidade e mais complexa do que a arquitetura VGG-16 original.

Inception

A rede neural Inception, muitas vezes referida como GoogLeNet (também descrita no item 2.3.2), é uma arquitetura de rede neural convolucional profunda projetada para tarefas de reconhecimento de imagem. Foi desenvolvida por pesquisadores do Google em 2014. A principal inovação da arquitetura Inception é o uso de módulos *inceptions*, que permitem à rede capturar recursos com eficiência em diferentes escalas e níveis de abstração.

A característica inovadora da arquitetura Inception é o módulo *inception*, o qual é composto por vários filtros convolucionais de diferentes tamanhos (por exemplo, 1x1, 3x3, 5x5) e camadas de *max pooling* em paralelo. Isso permite que a rede capture recursos em várias escalas espaciais e níveis de abstração dentro de uma única camada[65].

Inception V3 (2015) e Inception V4 (1016) representaram avanços na arquitetura Inception ao incorporar técnicas de outras arquiteturas de sucesso, enquanto mantiveram a ideia central de usar módulos *inception* para capturar recursos de forma eficiente em diferentes níveis de abstração.

Xception

Xception, abreviação de *Extreme Inception*, é uma arquitetura de rede neural convolucional (CNN) que se baseia nas ideias da arquitetura Inception original, mas com uma diferença: Xception emprega convoluções separáveis em profundidade para aumentar a eficiência e o desempenho. Ao usar convoluções separáveis em profundidade, o Xception aproveita o fato de que muitos recursos podem ser capturados com menos cálculos, tornando-o adequado para cenários com recursos computacionais limitados[87].

A concepção do Xception reside em substituir as camadas convolucionais padrão nas CNNs tradicionais por convoluções separadas, em profundidade. Essa modificação ar-

quitetônica reduz o número de parâmetros e cálculos, mantendo, ao mesmo tempo, a capacidade de representação das características da imagem.

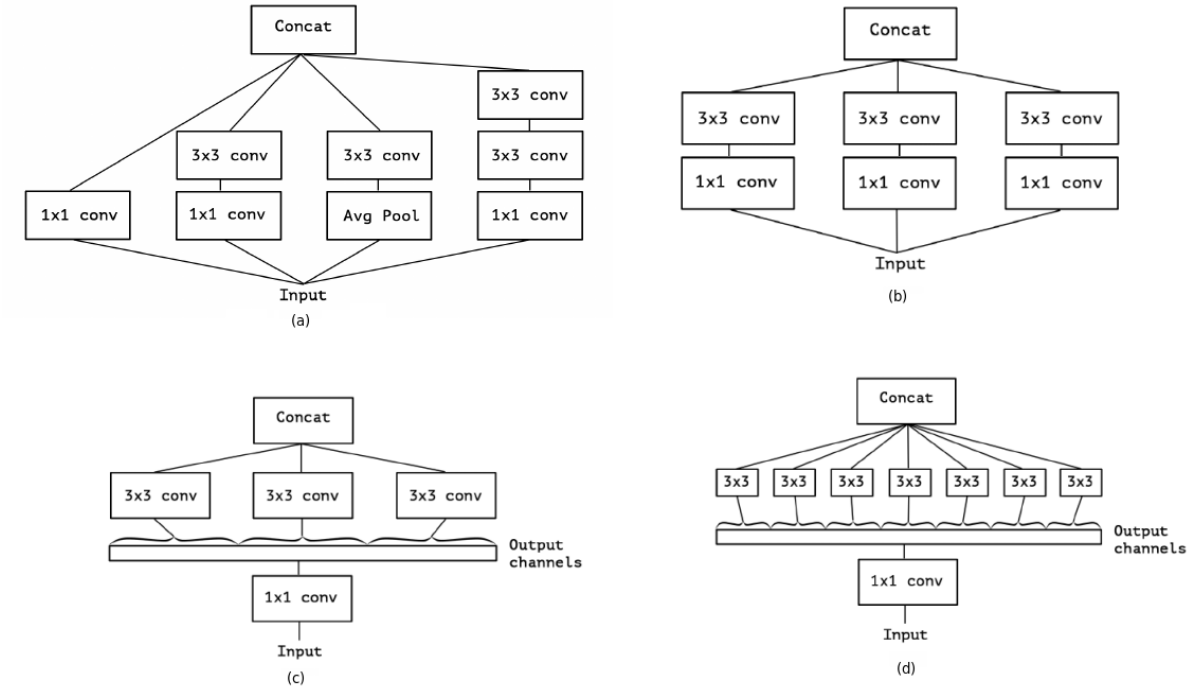


Figura 2.29: Comparativo da arquitetura Xception e Inception. Em (a) Um módulo Inception-V3; em (b) Uma simplificação do módulo Inception; em (c) Uma reformulação da simplificação do módulo Inception visto em (b); e em (d) A versão "Extrema"do módulo Inception

MobileNet

MobileNet é uma família de arquiteturas de redes neurais projetadas para computação eficiente no dispositivo, especialmente em dispositivos móveis e incorporados. Essas arquiteturas são otimizadas especificamente para alcançar um bom equilíbrio entre a precisão do modelo e a eficiência computacional, tornando-as adequadas para aplicações com recursos computacionais limitados. MobileNets foram introduzidas por pesquisadores do Google em 2017[88].

A rede MobileNet tem várias versões, incluindo a MobileNet-V1[88], a MobileNet-V2[89] (a qual é utilizada neste experimento) e a MobileNet-V3, cada um com suas próprias evoluções. A MobileNet-V2 introduziu blocos residuais invertidos, *skip connections* e *bottlenecks* lineares para melhorar a eficiência e a acurácia. A MobileNet-V3 otimizou ainda mais a arquitetura para diferentes aplicações e dispositivos.

Capítulo 3

Trabalhos Relacionados

Este capítulo traz uma revisão da literatura no que diz respeito aos estudos qualitativos com imagens degradadas e aprendizagem profunda. Inicialmente, buscou-se abordar somente os estudos com imagens degradadas para a tarefa de reconhecimento facial, entretanto, no decorrer da pesquisa, foram observados diversos outros estudos utilizando a mesma metodologia de degradação da imagem para outras tarefas. Por este motivo, alguns estudos que não abordam especificamente a tarefa de reconhecimento facial foram aqui relacionados.

Assim, a fim de melhor organizar os trabalhos relacionados ao tema principal da pesquisa proposta neste trabalho, dividimos os achados em três grupos aqui expostos:

1. Algoritmos de reconhecimento facial utilizando imagens da face degradadas;
2. Redes neurais convolucionais utilizando imagens degradadas; e
3. Algoritmos de avaliação da imagem da face utilizando imagens da face degradadas.

3.1 Reconhecimento Facial com Imagens Degradadas

De acordo com [78], foram avaliados três algoritmos de reconhecimento facial frente às distorções de borramento de movimento, ruído, compressão, distorções de cor e oclusões. Esta avaliação teve como foco identificar a influência das degradações no desempenho de cada algoritmo. Durante os experimentos, foram utilizados três algoritmos: AlexNet[9], VGG-Face[37] e GoogLeNet[8]; e, ao final, os resultados indicaram que as degradações de borramento, ruído e oclusão causaram uma significativa queda no desempenho dos algoritmos, ao passo que as distorções de cor (balanço e contraste) apresentaram resultados menos impactantes. Como fonte de dados para o experimento, o autor utilizou a base de dados LFW[35], a qual é usualmente utilizada como padrão de comparação de desempe-

nho entre modelos, entretanto, é considerada relativamente fácil para os algoritmos mais recentes.

Grm *et al.* [31] estudaram os efeitos das diferentes covariantes relacionadas à qualidade da imagem no reconhecimento facial. O estudo utilizou as degradações de ruído, borramento, ausência de *pixels*, brilho, compressão e cor em diferentes níveis de incidência. Ainda, o estudo foi realizado utilizando apenas a base LFW[35] e quatro algoritmos de reconhecimento facial, são eles: AlexNet[9], VGG-Face[37], GoogLeNet[8] e SqueezeNet[90]. Como resultado, o estudo verificou pontos fortes e fracos dos modelos utilizados, identificando que elevados níveis de ruído, borramento, ausência de *pixels* e baixo brilho prejudicam, de forma aguda, o desempenho dos algoritmos. Entretanto, modificações no contraste e compressões possuem uma influência mais branda.

3.2 Redes Neurais Convolucionais com Imagens Degradadas

Liu *et al.*[79] estudaram o desempenho de diferentes modelos CNNs, em diversas tarefas (reconhecimento e detecção facial, reconhecimento de objeto e classificação de imagens), em arquivos de imagem e vídeo. Durante a pesquisa, os autores utilizaram as seguintes degradações: redução da imagem em 4 vezes; ruído sal e pimenta; ruído gaussiano; borramento gaussiano; oclusões randômicas; redução da imagem em 4 vezes, seguido de ruído gaussiano; e redução da imagem em 4 vezes seguido de borramento gaussiano. Assim, observou-se que além de incluir arquivos de vídeo e tarefas como reconhecimento de objetos no estudo em questão, os autores incluíram também degradações em sequência no experimento, como, por exemplo, a redução da imagem seguido de ruído. Em relação às bases, devido à finalidade de cada tarefa, foram utilizadas bases com imagens de propósitos diferentes. Deste modo, a base de dados CIFAR-10[91] foi utilizada para reconhecimento dos objetos, a MSRA-CFW[92] para o reconhecimento facial, a FDDB[93] para a detecção facial, a SVHN[94] para o reconhecimento de dígitos e a ImageNet[51] para a classificação de imagens. Por fim, foi criado um modelo de rede neural profunda, utilizando transferência de conhecimento (do inglês *transfer learning*), com foco em imagens de baixa resolução.

Roy *et al.*[80] abordam o efeito das degradações das imagens em arquiteturas de redes neurais profundas para a tarefa de classificação de imagens. O trabalho avalia as degradações nos modelos CNNs, propõe novas configurações para melhora de desempenho. Ao final, os autores propõem uma CNN que atingiu melhores resultados em relação às que foram objeto de comparação no estudo. Durante o trabalho, foram utilizados para padrão de desempenho os modelos de CNN MobileNet[88][89], VGG16[10], VGG19[10],

ResNet50[7], InceptionV3[65] e, o modelo Capsulenet[80], o qual foi elaborado pelos autores. As imagens utilizadas no estudo foram degradadas utilizando ruído gaussiano, ruído sal e pimenta, borramento de movimento, borramento gaussiano, e compressão JPEG. Assim, após a execução do experimento, verificou-se que na degradação de ruído gaussiano, o desempenho das redes reduzem à medida que o ruído aumenta. Ainda sobre esta degradação, a arquitetura VGG se mostrou mais robusta em comparação às demais. No tocante à degradação de ruído sal e pimenta, todos os modelos foram afetados, em especial MobileNet[88][89]. Já na degradação de borramento de movimento e borramento gaussiano, todos os modelos tiveram seu desempenho degradado, entretanto, a arquitetura VGG[10] se mostrou mais consistente para a base de dados de dígitos. Por fim, a degradação de compressão JPEG não afetou o reconhecimento de qualquer modelo até determinada qualidade da compressão. Após este determinado ponto, ResNet[7] e MobileNet[88][89] decaíram de forma acentuada e arquitetura VGG[10] respondeu de maneira mais estável, apresentando os melhores resultados.

Estudos foram apresentados por Pei *et al.*[81] para verificar quanto o desempenho reduz ao recorrer a imagens degradadas para a tarefa de classificação de imagens. Ainda neste estudo, foi verificado se, ao incluir imagens degradadas na base de treinamento, o desempenho da rede poderia melhorar. Para este trabalho foram utilizadas imagens degradadas com ruído de névoa, brilho (escurecimento), borramento por movimento e lente olho de peixe. O estudo contou com imagens degradadas das bases de dados Caltech-256[95], PASCAL VOCs[96] e ImageNet[51] e, como espinha dorsal dos modelos, as redes CNNs AlexNet[9] e VGGNet-16[10]. Através dos resultados, os autores verificaram que o desempenho dos algoritmos de classificação de imagens reduziu, especialmente quando o treinamento da rede não pode refletir estes níveis de degradação das imagens utilizadas no teste, ou seja, quando na fase de treinamento são utilizadas apenas imagens de boa qualidade. Além disso, o estudo demonstrou que, ao visualizar as ativações das camadas escondidas das CNNs, diversas características importantes não foram discernidas, o que pode explicar o baixo desempenho das redes.

Aljarrah [82] procurou investigar como as degradações provocariam uma redução no desempenho de CNNs para a tarefa de classificação de imagens. Desta forma, o autor utilizou a rede GoogLeNet[8] com a popular base de imagens ImageNet[51]. Assim, as imagens foram degradadas por meio de borramento, contraste, ruído e oclusão, e, como resultado, verificou-se que o borramento produziu a queda mais acentuada de desempenho, seguido pela oclusão, e de maneira mais branda, o ruído. Neste estudo, o autor não utilizou imagens degradadas durante o treinamento, mas deixou claro que para trabalhos futuros estas imagens (degradadas) podem ser incluídas na base de treinamento, a fim de melhorar o desempenho da rede GoogLeNet.

3.3 Avaliação da Qualidade da Imagem com Imagens Degradadas

Singh *et al.*[97] realizam uma análise a respeito dos escores (do inglês *Quality Score* - QS) obtidos pelos algoritmos de avaliação de qualidade da face. Durante a análise, algumas degradações são realizadas nas imagens das faces para que os resultados possam ser avaliados a fim de verificar a robustez dos algoritmos. Durante o estudo, os autores utilizaram os algoritmos de qualidade da face FaceQNet[98], SER-FIQ[99], SDD-FIQA[100], BRUSQUE[101], PIQUE[102], NIQE[103], e, como base das imagens, foram selecionadas algumas imagens aleatoriamente das bases LFW[35] e IJB-C[104]. Após a produção dos resultados, os autores verificaram que o algoritmo SER-FIQ superou os demais na maioria dos experimentos, conseguindo apresentar uma curva de degradação (QS) de acordo com o nível das imagens degradadas apresentadas. Nesta análise, foram analisadas as degradações de ruído gaussiano, ruído sal e pimenta, ruído de manchas, oclusões, brilho (clareamento e escurecimento), bordas coloridas e compressão JPEG.

3.4 Avaliação dos Trabalhos Relacionados

De forma geral, os estudos citados utilizaram algoritmos de reconhecimento facial com desempenho inferiores aos algoritmos disponíveis atualmente, provavelmente devido ao momento em que os trabalhos foram elaborados e publicados. Desta forma, verifica-se que um estudo com algoritmos mais atualizados se mostra válido.

Ainda, percebe-se que grande parte dos trabalhos referentes à tarefa de reconhecimento facial utilizaram apenas uma base de imagens para processar os experimentos, neste ponto, acredita-se que isso se deve ao alto custo de *hardware* e ao tempo de processamento necessário que este estudo exige. Não foram observadas análises dos algoritmos de reconhecimento facial juntamente com os de detecção facial no mesmo contexto, de forma que pudéssemos observar qual combinação de algoritmos apresenta um desempenho mais consistente frente às degradações apresentadas.

Por fim, quanto aos algoritmos de avaliação da qualidade da face, verifica-se que atualmente existem soluções mais modernas, com resultados mais robustos. Desta forma, assim como mencionado acerca dos algoritmos de reconhecimento facial, um estudo mais recente com tais algoritmos também se mostra pertinente.

Para uma melhor visualização dos trabalhos aqui descritos, bem como demonstrar as diferenças com a proposta atual, elaboramos a Tabela 3.1, contendo as principais informações.

Trabalho	**Modelos**	**Degradações**	**Datasets**
[78]	AlexNet VGG-Face GoogLeNet	Borramento de movimento, Distorções de cor Ruído Compressão, Oclusões	LFW (RecFac)
[31]	AlexNet VGG-Face GoogLeNet	Ruído Borramento Ausência de pixels, Compressão, Variações de cor	LFW (RecFac)
[79]	Modelo desenvolvido pelo autor	Redução de imagem em 4x, Sal e pimenta, Ruído gaussiano, Borramento gaussiano Oclusões randômicas, Redução da imagem em 4x + ruído gaussiano, Redução da imagem em 4x + borr. gaussiano	CIFAR-10 (DetObj), MSRA-CFE (RecFac), FDDB (DetFac), SVHN (RecDig), ImageNet (ClsImg)
[80]	MobileNet VG-16 VGG-19 ResNet-50 Inception-v3 Capsulenet	Ruído gaussiano Sal e pimenta Borramento de movimento Borramento gaussiano Compressão JPEG	
[81]	AlexNet GoogLeNet	Ruído de névoa, Brilho, Escurecimento, Borramento de movimento, Lente olho de peixe	Caltech-256 (ClsImg), PASCAL VOC (ClsImg), ImageNet (ClsImg)
[82]	GoogLeNet	Borramento, Contraste, Ruído, Oclusão	ImageNet (ClsImg)

Tabela 3.1: Resumo dos trabalhos abordados nesta seção

Capítulo 4

Metodologia Proposta

Este capítulo descreve os métodos de trabalho que serão utilizados na pesquisa. Para atingir os objetivos propostos na Seção 1.1, utilizaremos uma metodologia experimental, dividida em duas fases, que serão desenvolvidas e executadas de forma sequencial.

Na seção a seguir, descrevemos cada uma das fases separadamente.

4.1 Experimento Proposto

Para atingir o objetivo geral proposto, ou seja, quantificar, comparar, consolidar e estimar o impacto da degradação da imagem no desempenho dos sistemas de reconhecimento facial, o projeto foi dividido em duas fases, conforme Figura 4.1, sendo:

Fase 1 - Avaliação da Influência das Degradações em Modelos Profundos: Quantificar, comparar e consolidar o impacto das degradações nos sistemas de reconhecimento facial utilizando modelos de aprendizagem profunda, utilizando imagens degradadas de diversas bases de dados públicas;

Fase 2 - Definição de Modelos para Detecção de Degradações: Identificar o tipo de degradação e a intensidade presente em uma imagem fazendo uso de um modelo de aprendizado profundo e estimar a perda de desempenho de determinado sistema de reconhecimento facial utilizando a base de dados coletada na fase anterior.

4.1.1 Fase 1 - Avaliação da Influência das Degradações em Modelos Profundos

Para organizar o fluxo de trabalho proposto, a Fase 1 foi subdividida em 4 etapas, conforme descrito abaixo e exibido na Figura 4.2:

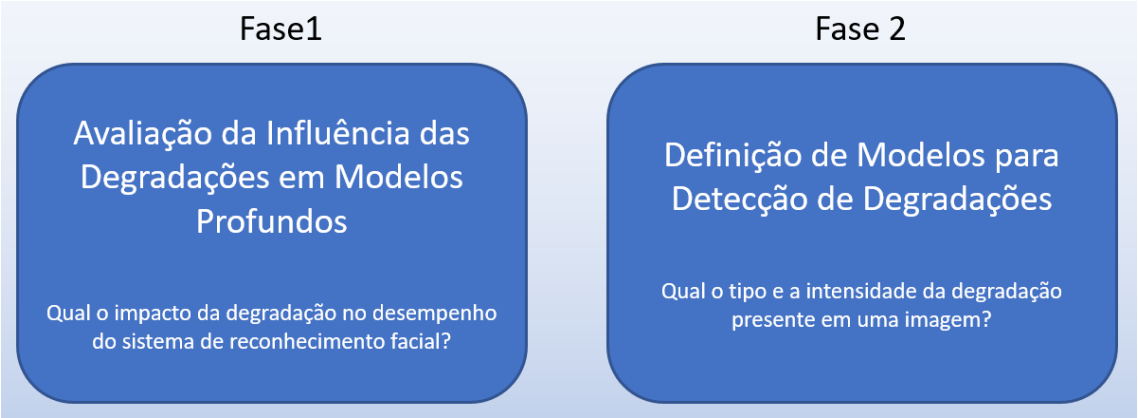


Figura 4.1: Fases da metodologia proposta.

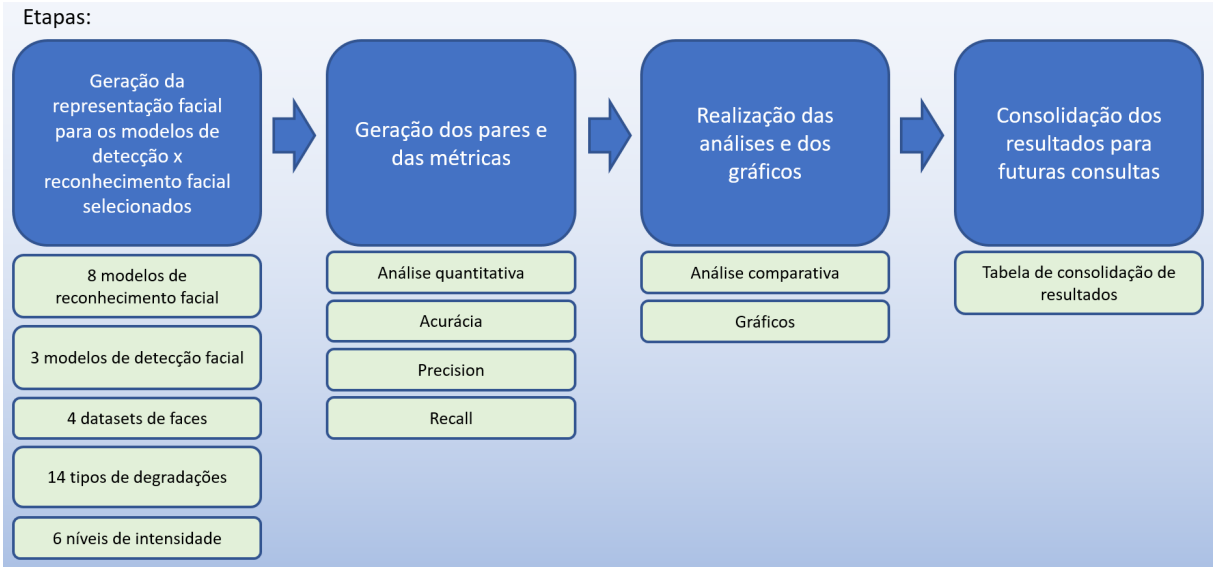


Figura 4.2: Descrição das Etapas da Fase 1.

Etapa 1 - Geração das representações faciais: Geração das representações com as imagens originais e degradadas, utilizando 4 (quatro) conjunto de imagens, 14 (quatorze) degradações com 6 (seis) níveis de intensidade em cada degradação;

Etapa 2 - Geração dos pares e das métricas: Gerar os pares de imagens para o cálculo da similaridade e, em seguida, das métricas utilizadas na proposta;

Etapa 3 - Geração de gráficos: Geração de gráficos para realização de análises comparativas a fim de detectar novas tendências e descobertas acerca do desempenho dos sistemas de reconhecimento facial diante de imagens degradadas;

Etapa 4 - Consolidação dos resultados: Consolidar os resultados obtidos nas etapas anteriores, de forma que sejam disponibilizados para futuras consultas de predições.

A fim de medir o impacto das degradações frequentemente observadas no desempenho dos sistemas de reconhecimento facial, esta proposta de experimento utilizará diversos algoritmos para a geração da representação da face. Os modelos utilizados são considerados estado da arte, foram previamente treinados, e fazem parte da biblioteca denominada DeepFace[105], a qual encontra-se disponível gratuitamente no repositório da linguagem Python. O experimento proposto usará 8 (oito) modelos de reconhecimento facial: VGG-Face[**?**], FaceNet[38], OpenFace[39], DeepFace[40], DeepID[41], ArcFace[17], Dlib[14] e SFace[18].

Para a detecção e o alinhamento da face, e a fim de trazer maior robustez ao experimento proposto, também serão utilizados algoritmos amplamente reconhecidos e utilizados. Os algoritmos de detecção também foram previamente treinados e estão integrados à biblioteca DeepFace, citada anteriormente. Desta forma, serão utilizados 3 (três) algoritmos para detecção das faces: Dlib[14], MTCNN[43], RetinaFace[16].

As bases de dados (imagens de face) utilizadas estão disponíveis publicamente, e serão submetidas ao processo de degradação de imagem. Estas degradações variaram em relação ao tipo e em relação ao fator da intensidade (da degradação) sobre a imagem. Serão utilizados 14 (quatroze) tipos de degradações, sendo 8 (oito) degradações puras (ou seja, apenas um tipo de degradação) e 6 (seis) degradações combinadas em sequência (conjunto de degradações aplicadas em sequência na imagem). Para cada tipo de degradação, cada imagem foi degradada sob 6 (seis) níveis de intensidade diferentes, de forma gradual, assim, poderá ser avaliado a curva de desempenho e identificar, a partir de qual determinado nível de degradação, o sistema de reconhecimento facial eventualmente começará a falhar.

Determinar as degradações utilizadas neste estudo foi uma tarefa árdua, tendo em vista a falta de material disponível neste contexto. Ainda, uma vez que a pesquisa foca em dados utilizados em exames periciais, a escassez de material se torna ainda mais

evidente. Desta forma, buscamos outras fontes oficiais para a escolha das degradações aqui utilizadas. A organização ISO (*International Organization for Standardization*), a qual busca estabelecer padrões para diversas áreas de conhecimento, mantém um documento do qual trata da qualidade das imagens em sistemas biométricos (ISO/IEC 19794-5:2011)[32], e de forma mais específica, da qualidade das imagens utilizadas no reconhecimento facial (ISO/IEC TR 29794-5:2010)[75].

Cabe ressaltar que o referido documento que trata de imagens utilizadas no reconhecimento facial (ISO/IEC TR 29794-5:2010) está em revisão neste momento, sendo possível analisar apenas a sua nova versão de rascunho (ISO/IEC WD 29794-5:2022 - *Working Draft (WD)*)[75]. Apesar de ainda ser um rascunho, o documento permite realizar analises no tocante às degradações que podem ser utilizadas, de forma que mesmo com eventuais mudanças (por ainda ser um rascunho), a maioria dos estudos aqui realizados ainda sejam relevantes.

O referido documento (ISO/IEC WD 29794-5:2022)[75], no Item 6, Subitem 6.3, denominado Elementos de Qualidade relacionados à Captura, possui 14 (quatorze) itens a serem considerados, destes, 8 (oito) itens serão abordados diretamente por esta pesquisa e, os outros 6 (seis) itens serão abordados indiretamente, através das características dos conjuntos de dados escolhidos. Complementando, ainda no Item 6, Subitem 6.4, denominado Elementos de Qualidade relacionados ao Sujeito, o documento possui 10 (dez) itens, destes 4 (quatro) serão abordados indiretamente, através dos conjuntos de dados. Assim, apesar de não cobrir todos os tópicos do referido documento, buscamos contemplar o máximo de itens possível.

Ainda no tocante à decisão em relação às degradações utilizadas neste trabalho, vale dizer que as degradações puras foram, de forma geral, utilizadas em pesquisas anteriores[78, 31, 79, 80, 81, 82], já que são degradações costumeiramente analisadas neste tipo de trabalho. Entretanto, as degradações em sequência, bem mais raras nos trabalhos científicos, foram combinadas de forma que fosse possível representar os problemas reais encontrados no dia a dia na atividade pericial.

Por exemplo, a degradação combinada em sequência Borramento Gaussiano $\rightarrow$ Escurecimento $\rightarrow$ Redução de Tamanho $\rightarrow$ Compressão JPEG, busca representar um cenário extremamente reincidente nos exames, o qual o suspeito encontra-se longe do ponto de captura da câmera, na via pública e à noite. Somando-se a isso, um equipamento de baixa qualidade usualmente encontrado nas residências e um algoritmo de compressão com perdas, que também é usualmente empregado nestes arquivos de imagens, seja para armazenamento ou transmissão dos dados.

Etapa 1 da Fase 1

O processamento de cálculo de representação é custoso, pois demanda equipamento (servidores com placas gráficas) e tempo de processamento. Nesta etapa da Fase 1, o objetivo é a geração dos vetores de representação de cada imagem (incluindo a original e as 6 degradadas) e seu armazenamento em arquivos no formato *Comma-Separated Values* (csv). Optou-se por gravar os resultados desta etapa em arquivos no formato csv para preservar o processamento das representações e utilizar os dados para o cálculo de métricas e realização de análises, evitando, assim, reprocessamento do cálculo da representação da mesma imagem.

Para cada imagem original oriunda das bases de dados, serão geradas, em tempo de execução, 6 (seis) imagens degradadas, onde a intensidade da degradação será aumentada de forma gradual. A tabela 4.1 expõe os parâmetros utilizados na geração das degradações.

Degradação	**Nível 1**	**Nível 2**	**Nível 3**	**Nível 4**	**Nível 5**	**Nível 6**	**Fator**
Bor. Gaussiano	3	7	11	15	19	23	tam. janela
Bor. Movimento	3	5	7	9	11	13	tam. janela
Brilho	(1.0, 1.0)	(1.0, 1.5)	(1.0, 2)	(1.0, 2.5)	(1.0, 3)	(1.0, 3.5)	(alpha, beta)
Redução de Tam.	85	70	55	40	25	10	escala de tam.
Escurecimento	0.9	0.7	0.5	0.4	0.3	0.2	gamma
Compress. JPEG	85	70	55	30	15	5	escala qualidade
Ruído Gaussiano	0.001	0.01	0.03	0.07	0.1	0.15	escala ruído
Sal e Pimenta	0.001	0.01	0.03	0.07	0.1	0.15	escala ruído

Tabela 4.1: Níveis de degradação.

Para as degradações em sequência, serão aplicadas uma combinação de parâmetros conforme o nível da degradação. Ou seja, serão utilizados os mesmos parâmetros das degradações "puras", aplicados conforme o nível de severidade da degradação. Por exemplo, na geração da primeira imagem degradada da degradação Borramento Gaussiano → Compressão JPEG serão utilizados os primeiros parâmetros de cada degradação pura, ou seja, para o borramento, o filtro gaussiano de tamanho do *kernel* 3 e, para a compressão, qualidade da imagem de 85%. Na geração da segunda imagem degradada, serão utilizados os segundos parâmetros de cada degradação pura, ou seja, *kernel* igual a 7 e qualidade da imagem de 70%. As demais imagens das degradações em sequência serão geradas utilizando o mesmo raciocínio.

Em relação à degradação do Brilho, os parâmetros *alpha* e *beta*, são equivalentes ao coeficiente de escala e coeficiente de adição. Assim, utilizamos uma função que multiplica cada *pixel* da matriz de entrada (imagem) pelo coeficiente de escala (*alpha*) e, em seguida, adicionamos o coeficiente de adição (*beta*). Os valores resultantes são convertidos para valores absolutos, garantindo que não haja números negativos na matriz resultante.

O parâmetro *gamma* da degradação Escurecimento é um fator aplicado para clarear ou escurecer uma imagem. Neste caso específico, será empregado no intuito de escurecer a imagem. Assim, após a imagem ser transformada para escala de 0-1, aplicaremos a fórmula a seguir, e depois retornaremos para escala de 0-255, como mostrado na Equação 4.1.

$$\mathrm{p} = \mathrm{o} * \frac{1}{\mathrm{g}} \tag{4.1}$$

onde p significa Imagem Processada, o significa Imagem Original e g, gamma.

As degradações foram realizadas para uma imagem presente na base de dados LFW[35], e foram exibidas nas Figuras 2.26 e 2.27. Na figura 2.26, são exibidas as degradações puras, e na Figura 2.27, as degradações em sequência.

Nesta etapa, cabe dizer que um cenário passível de ser encontrado é o que denominamos de "Problema da quebra do *pipeline*". Este problema ocorreria quando a degradação é tão intensa na imagem, que o algoritmo de detecção facial não consegue mais detectar uma face e, consequentemente, o *pipeline* é interrompido. O fato interessante neste caso é que como temos diferentes algoritmos de detecção facial, com diferentes acurácias e níveis de robustez frente às degradações inseridas, essa interrupção poderia ser causada em momentos distintos entre eles.

Desta forma, a fim de preservar a equidade no experimento e realizar de forma justa as comparações entre todos os *pipelines* estudados, foi decidido por apenas capturar o momento da quebra do pipeline, mas seguir com o processo até o final, ou seja, mesmo que o algoritmo de detecção facial não encontre uma face, passamos a imagem para algoritmo de reconhecimento facial gerar as representações e, posteriormente, fazer a geração dos pares e os cálculos de similaridade entre pessoas.

Essa foi a maneira encontrada de manter a igualdade de procedimento necessária para avaliar todos os algoritmos, tanto os de detecção quanto os de reconhecimento, já que todos são submetidos exatamente ao mesmo processo. A única alteração no fluxo acontece quando ocorre a quebra, neste caso, nós apenas capturamos o momento para, posteriormente, realizar a análise de desempenho dos algoritmos de detecção. Em todos os casos, havendo quebra ou não, o *pipeline* segue até o final.

Importante ressaltar que para retratar cenários mais próximos da realidade, os conjuntos de dados capturados em ambiente controlado serão pré-processados. Dado que os conjuntos de dados SCFace[84], FEI[83] e GUFD[85] contêm rostos capturados em um ambiente controlado, onde são capturados vários ângulos do rosto de uma pessoa (variando de 90 graus para a esquerda a 90 graus para a direita), e para manter apenas o rosto frontal imagens dos conjuntos de dados, os conjuntos de dados mencionados foram submetidos ao detector facial frontal da biblioteca Dlib. Desta forma, uma "versão limpa" dos

conjuntos de dados será gerada e utilizada durante o experimento proposto. A Tabela 4.2 apresenta o número real de imagens que serão utilizadas, para cada conjunto de dados, na abordagem proposta.

Conjunto de dados	**Total de imagens no conjunto de dados original**	**Total de imagens utilizadas**	**Total de classes**
LFW	13.233	13.233	5.749
SCFace	1.170	833	130
FEI	2.800	2.450	200
GUFD	3.028	1.747	304

Tabela 4.2: Versão limpa dos conjuntos de dados escolhidos.

Etapa 2 da Fase 1

A geração dos pares se dará pela combinação de imagens de pares positivos e negativos, ou seja, pares de imagens de faces que são da mesma pessoa e de pessoas diferentes. Ainda, serão combinadas imagens degradadas e não degradadas. Os pares a serem estabelecidos serão descritos em detalhes a seguir.

Para simular diferentes cenários encontrados com frequência, serão gerados três tipos diferentes de pares de comparação, como segue: Par 1 - Imagem padrão (não degradada) vs. Cópia da imagem padrão (degradada); Par 2 - Imagem padrão (não degradada) vs. Imagem questionada (degradada); Par 3 - Imagem padrão (degradada) vs. Imagem questionada (degradada).

Estes cenários podem ser observados no documento intitulado *FISWG FR Systems Methods and Techniques - version 1.0*[34], onde são descritos diversas possibilidades de comparação entre imagens de faces, variando a qualidade entre alta e baixa. Aqui, no caso concreto, a pesquisa busca contribuir neste tema explorando diferentes tipos de degradações, intensificando gradualmente a degradação inserida na imagem.

A seguir, descrevemos em detalhes a geração dos pares:

Par 1 - Imagem padrão (não degradada) x Cópia da imagem padrão (degradada): Degradações (em seis níveis de intensidade) serão aplicadas a uma cópia da imagem padrão e depois serão submetidas ao sistema de reconhecimento facial. O objetivo de gerar este par é determinar o nível de intensidade de degradação que o algoritmo mantém para reconhecer o par de imagens idênticas (uma não degradada e outra degradada) com a mesma pessoa. Então, por se tratar de uma cópia da própria imagem analisada, não existem pares negativos nesta geração;

Par 2 - Imagem padrão (não degradada) x Imagem questionada (degradada): Degradações (em seis níveis de intensidade) serão aplicadas a todas as imagens questionadas e submetidas a sistemas de reconhecimento facial. O objetivo de gerar este par é simular o uso do sistema quando uma imagem é de qualidade relativamente boa, como imagens de documentos oficiais, enquanto a outra imagem é obtida em um ambiente não controlado e com degradações aplicadas a ela;

Par 3 - Imagem padrão (degradada) x Imagem questionada (degradada): Nesta geração de pares, as degradações serão aplicadas a ambas as imagens. O objetivo é examinar casos em que ambas as imagens foram obtidas em ambiente não controlado (como imagens de redes sociais ou locais públicos) e avaliar o desempenho do algoritmo nesses cenários;

Em seguida, após a geração dos pares, serão calculadas as distâncias de similaridade de cosseno dos pares, utilizando as representações já calculadas na etapa anterior. A fim de determinar se um par é da mesma pessoa ou de pessoas diferentes, serão utilizados os limiares padrões indicados, para cada modelo, na própria biblioteca DeepFace[105], a qual disponibiliza os modelos ora utilizados.

Desta forma, após o cálculo das similaridades e a predição se o par de imagens apresenta a mesma pessoa ou pessoas diferentes, serão utilizadas métricas comumente aplicadas em projetos de aprendizado profundo - acurácia, precisão e revocação, conforme descritas no Item 2.7.

Figura 4.3: Esquema da geração dos pares de imagens.

Etapa 3 da Fase 1

Após a geração das métricas de desempenho, serão criados gráficos para facilitar a visualização do desempenho obtido, bem como comparar os desempenhos entre os algoritmos e, eventualmente, identificar novas tendências/descobertas a respeito da robustez dos algoritmos. Nesta etapa foi possível analisar, de forma comparativa, a eficiência das diversas combinações de algoritmos utilizados no fluxo do sistema de reconhecimento facial.

Etapa 4 da Fase 1

Como etapa final da Fase 1 da proposta, será realizada uma sumarização detalhada (tabela consolidada) com as informações a respeito dos algoritmos de reconhecimento, algoritmos de detecção, tipos e intensidades das degradações, e as métricas obtidas. O objetivo desta tabela é servir de guia de referência para futuras predições, permitindo estimar qual o percentual de perda de desempenho determinado algoritmo pode eventualmente ter a partir das imagens utilizadas como entrada.

4.1.2 Fase 2 - Definição de Modelos para Detecção de Degradações

Para atingir o objetivo de identificar a eventual degradação (e a intensidade) presente na imagem de um rosto, será utilizado um conjunto de imagens faciais comumente usado como referência para aplicações de reconhecimento facial: *Labeled Faces in the Wild* (LFW) como fonte original para criar um conjunto de dados de imagens degradadas, que serviu de base para o treinamento. Este conjunto de dados foi degradado em 14 (quatorze) tipos diferentes de degradações, incluindo degradações puras e degradações sequenciais, em 6 (seis) níveis diferentes de intensidade.

Posteriormente, serão selecionadas e treinadas 10 (dez) arquiteturas consideradas estado da arte, tanto do zero (do inglês *from scratch*) quanto com técnica de transferência de aprendizado (do inglês *transfer learning*). Em outras palavras, um total de 20 modelos diferentes de CNN serão treinados no conjunto de dados mencionado acima. Para manter a equidade e permitir comparações, o treinamento de todos os modelos será realizado com os mesmos hiperparâmetros e pela mesma quantidade de épocas.

Assim como a fase anterior, a Fase 2 foi subdividida para melhor organização da metodologia proposta. Desta forma, a fase foi subdividida em 2 Etapas, conforme descrito abaixo e exibido na Figura 4.4:

Etapa 1) - Criação das bases de Imagens Faciais Degradadas: Criar as bases de treinamento e teste, utilizando imagens degradadas da base pública LFW, utilizada na Fase 1;

Etapa 2) - Treinamento, Evolução e Teste dos modelos: Treinar, evoluir e testar os modelos.

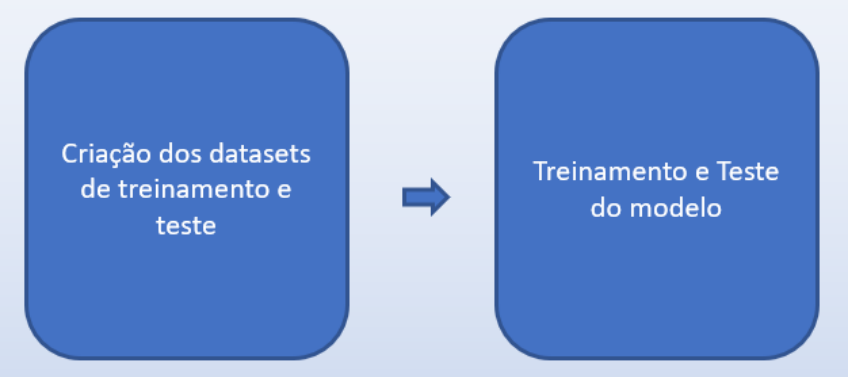


Figura 4.4: Fase 2

Etapa 1 da Fase 2

A Etapa 1 se caracteriza pela construção dos conjuntos de imagens faciais degradadas de treino e de teste, de forma a permitir avaliar e definir qual o melhor modelo a ser utilizado para esta tarefa. Como base do conjunto de imagens utilizado durante a fase de treinamento e teste, será utilizado o conjunto de dados Labeled Faces in the Wild (LFW), o qual já foi descrito neste trabalho, no subitem 2.11.1.

Para construir um conjunto de dados de treinamento e teste, o conjunto de imagens faciais da LFW será submetido à degradação em 14 (quatorze) tipos distintos de degradações (conforme exposto no item da Seção 2.10), em 6 (seis) níveis de intensidade diferentes (Tabela 4.1), resultando na criação de 84 classes de imagens. Além disso, uma classe contendo as imagens originais (não degradadas) será incorporada a este conjunto de dados. Consequentemente, o conjunto de dados empregado para o experimento compreenderá um total de 85 classes (14 tipos de degradação x 6 intensidades + 1 classe para imagens não degradadas). Considerando cada classe contendo 13.233 imagens, o conjunto de dados resultante abrangerá um total de 1.124.805 imagens.

Dessa contagem total de imagens, separadas de forma randômica, 70% das imagens serão utilizadas para o treinamento dos modelos, enquanto os 30% restantes serão alocados para fins de teste. Todas as estimativas e informações da quantidade de dados que serão utilizadas nesta fase estão resumidas na Tabela 4.3

Total de Classes	85
Total de Imagens por Classe	13.233
Total de Imagens no Conjunto de Dados	1.124.805
Total de Imagens de Treinamento	787.364
Total de Imagens de Teste	337.441

Tabela 4.3: Informações dos *Datasets*

Etapa 2 da Fase 2

O objetivo desta etapa é o treinamento, evolução e teste dos modelos para detecção de degradações nas imagens.

Para atingir um resultado satisfatório, e comparar desempenhos entre diferentes arquiteturas, foram selecionadas 10 (dez) arquiteturas consideradas estado da arte. Estas 10 (dez) arquiteturas escolhidas (ResNet-18, ResNet-50, ResNet-152, DenseNet-121, DenseNet-201, VGG-19, Inception-v3, Inception-v4, Xception-71, MobileNet-v2-100) serão treinadas na abordagem *from scratch* e com *transfer learning*.

Na primeira abordagem, os parâmetros iniciais da arquitetura foram inicializados aleatoriamente, enquanto na segunda, os parâmetros iniciais utilizados foram referentes ao treinamento do modelo equivalente, com base no conjunto de dados do ImageNet. O objetivo da segunda abordagem é justamente aproveitar o treinamento realizado anteriormente, com outro conjunto de dados, a fim de solucionar outro problema, e adaptar os pesos para a resolução de um problema específico, no caso deste trabalho, a classificação de 85 classes (tipos/intensidades de degradações).

O objetivo da escolha destas abordagens de treinamento é avaliar a resposta que os modelos geram quando possuem os pesos pré-carregados e quando são inicializados aleatoriamente. O resultado deste estudo será apresentado no capítulo seguinte.

ImageNet[51] é um grande conjunto de dados, diversamente utilizado para treinamento de aplicações de aprendizado profundo. Devido a sua quantidade de imagens e variedade de objetos, pode facilitar pesquisas em reconhecimento de imagem, detecção de objetos e classificação de imagem.

Por fim, cabe reforçar que a metodologia proposta neste capítulo será validada/confirmada no Capítulo 5 (Resultados).

Capítulo 5

Resultados

Este capítulo apresenta os resultados obtidos na pesquisa e foi organizado de acordo com as Fases do projeto, descritos no Capítulo 4. Desta forma, inicialmente apresentaremos os resultados da Fase 1 - Avaliação da Influência das Degradações em Modelos Profundos e, posteriormente, os resultados da Fase 2 - Definição de Modelos para Detecção de Degradações.

5.1 Materiais Utilizados

Para a realização do treinamento dos modelos, foram utilizados o framework PyTorch combinado com a biblioteca Fast.ai. Ainda, dado que foram utilizados servidores de uso coletivo, para a execução das rotinas, foram utilizados soluções de Docker, a fim de segmentar e proteger o *root* do sistema operacional entre os usuários. Os experimentos foram divididos e processados em 4 GPUs Nvidia Tesla V100.

5.2 Resultados da Fase 1 - Avaliação da Influência das Degradações em Modelos Profundos

Por se tratar de uma pesquisa extensa, onde foram gerados muitos resultados, para diferentes conjunto de imagens, preferimos expor aqui, apenas alguns resultados que embasam análises mais relevantes. Todos os resultados estão expostos, na íntegra, no Anexo 1.

5.2.1 Gráficos de Impacto dos Modelos de Reconhecimento Facial

Para exibir e analisar os resultados obtidos na Fase 1 do projeto, ou seja, geração das bases de dados referentes ao impacto da degradação nos sistemas de reconhecimento facial, escolhemos o algoritmo VGGFace, que conforme será exposto, obteve os melhores escores empregado com as imagens do conjunto de imagens da LFW. Todos os experimentos são direcionados para observar e quantificar os efeitos de imagens degradadas em sistemas de reconhecimento facial.

Para melhor visualização, os nomes das degradações avaliadas foram abreviados como segue: Borramento Gaussiano (GBlur), Borramento de Movimento (MBlur), Brilho (Brilho), Redução de tamanho (Down), Escurecimento (Dark), Compressão JPEG (JPEG), Ruído Gaussiano (GNoise), e, Sal e Pimenta (SP).

Os resultados dos impactos foram exibidos na forma de gráficos de linhas, onde no eixo x está a métrica revocação, e, no eixo y, a intensidade da degradação.

O Par 1 - Imagem padrão (não degradada) x Cópia da imagem padrão pode ser observado na Figura I.8. O Par 2 - Imagem padrão (não degradada) x Imagem questionada (degradada) na Figura I.16 e o Par 3 - Imagem padrão (degradada) x Imagem questionada (degradada) na Figura I.24.

5.2.2 Análises dos Algoritmos de Reconhecimento Racial

A análise foi organizada a partir da observação, considerando todos os resultados do Capítulo I - Anexo 1, de duas perspectivas: geração de pares e perspectiva do modelo, divididas por *datasets*.

Análise de geração de pares: Examinamos o processo de geração de pares de faces para a tarefa de reconhecimento e detecção. Avaliamos a eficácia da técnica de geração de pares na captura de características e variações faciais relevantes;

Perspectiva do modelo : Nosso foco é avaliar o desempenho de diferentes modelos de reconhecimento e detecção facial. Analisamos o desempenho desses modelos em vários conjuntos de dados, destacando seus pontos fortes e fracos, subdivididos por conjuntos de dados.

Análise de Geração de Pares

Ao analisar os resultados, observou-se que a métrica revocação representou melhor o estudo. Considerando que o número de pares negativos é muito superior ao número de

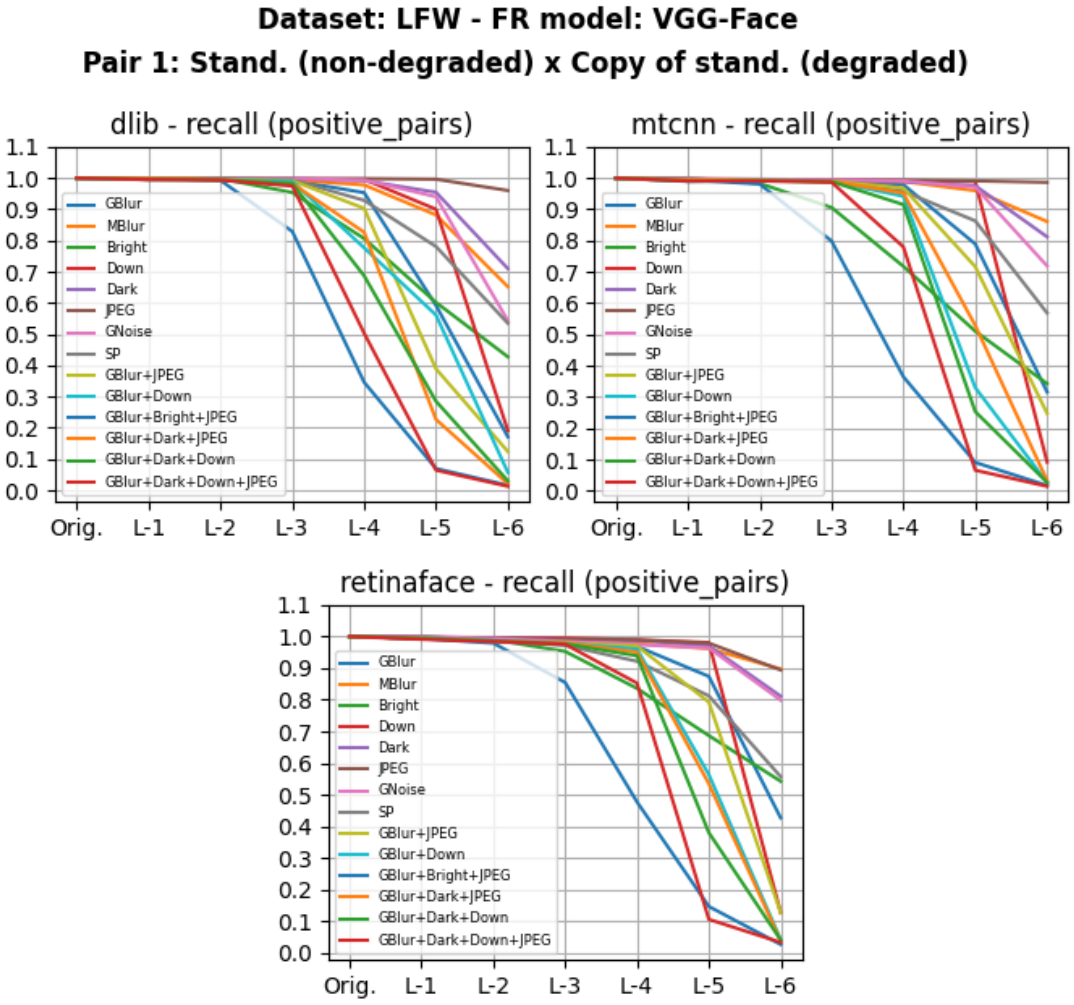


Figura 5.1: Métrica revocação do algoritmo VGG.

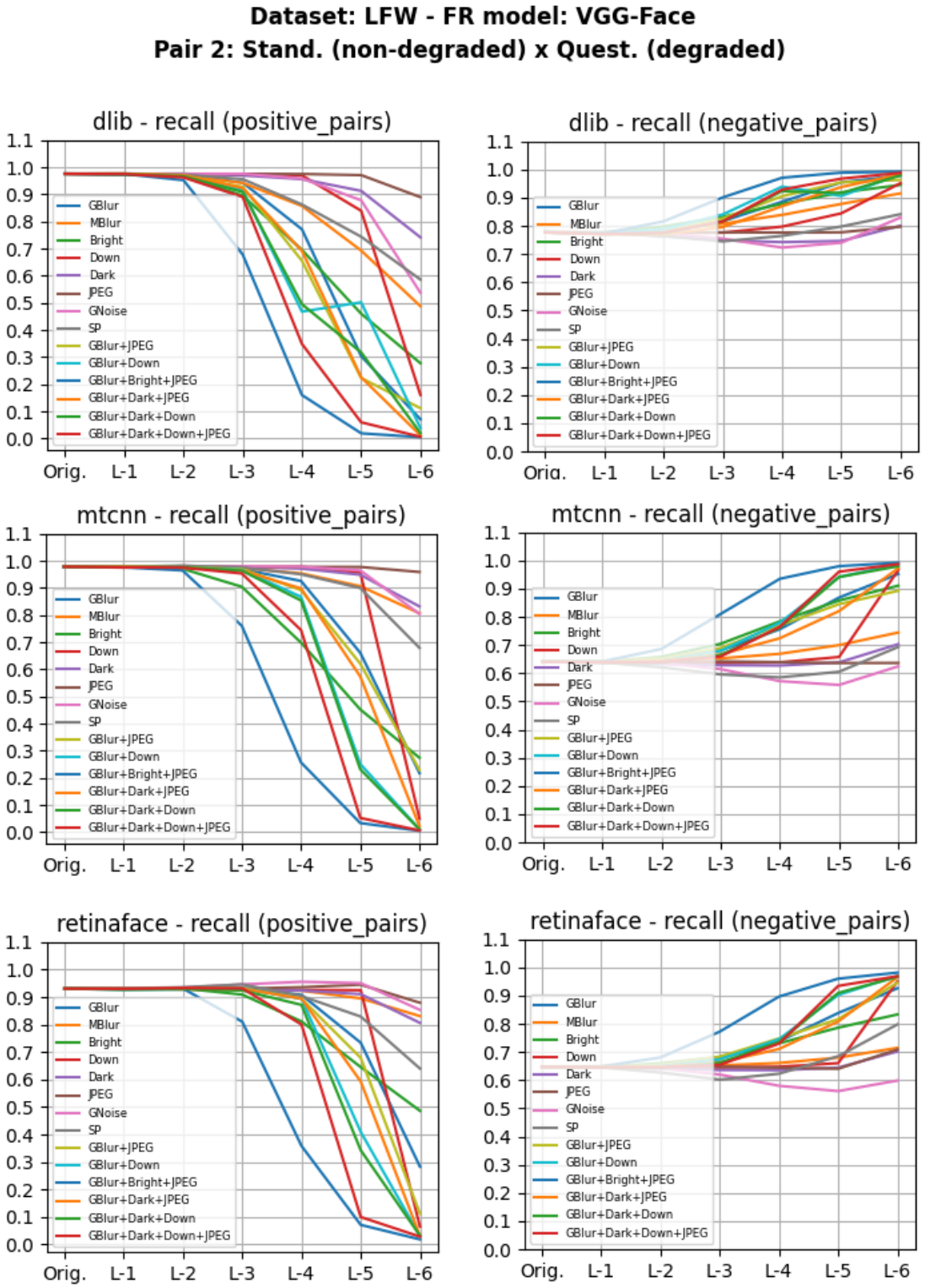


Figura 5.2: Métrica revocação do algoritmo VGG.

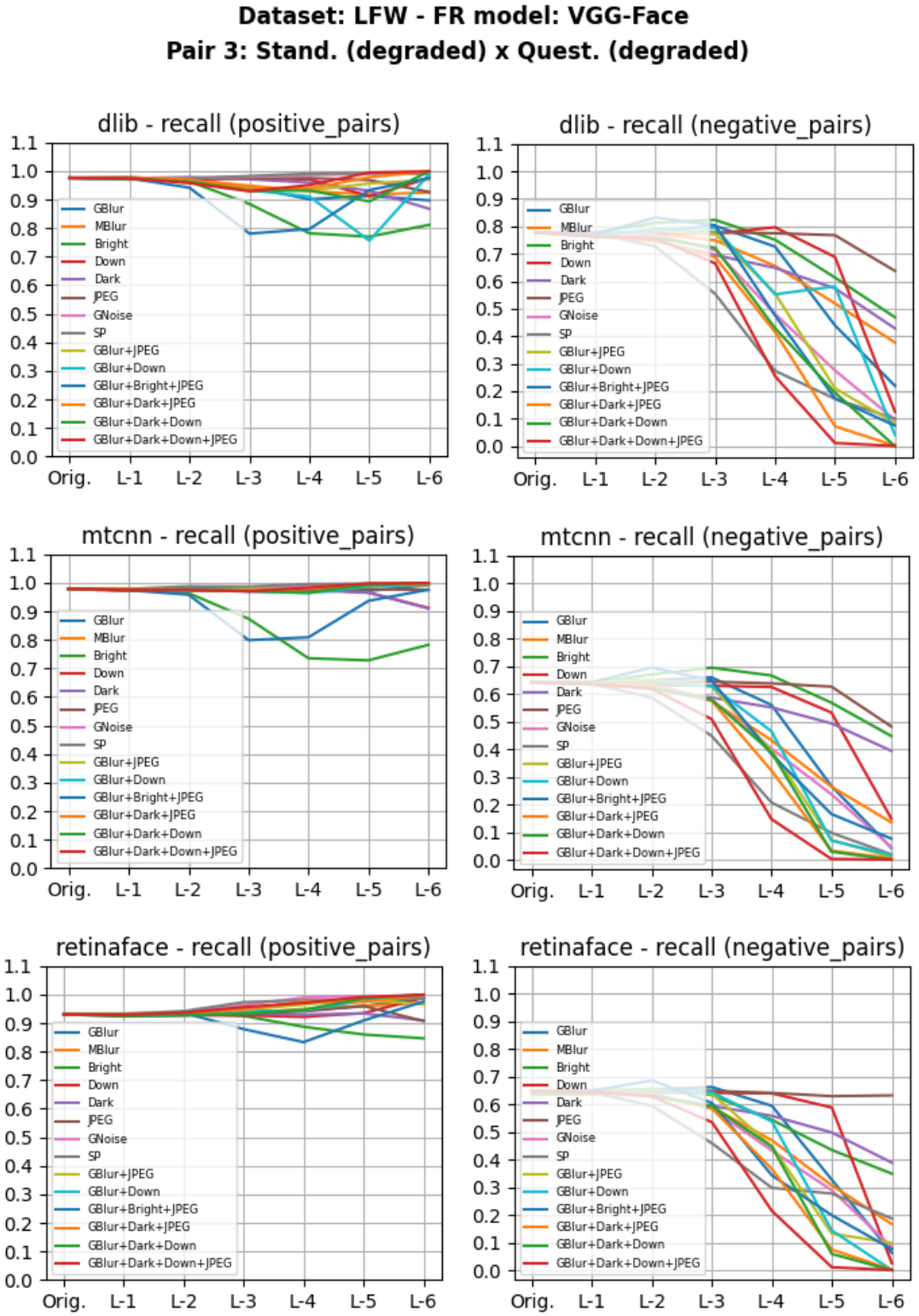


Figura 5.3: Métrica revocação do algoritmo VGG.

pares positivos, a métrica de acurácia acabou seguindo a tendência da curva de pares negativos.

Esta percepção pode ser visualizada através da Figura 5.4, onde é possível ver a métrica da acurácia geral seguindo a tendência da métrica revocação para pares negativos.

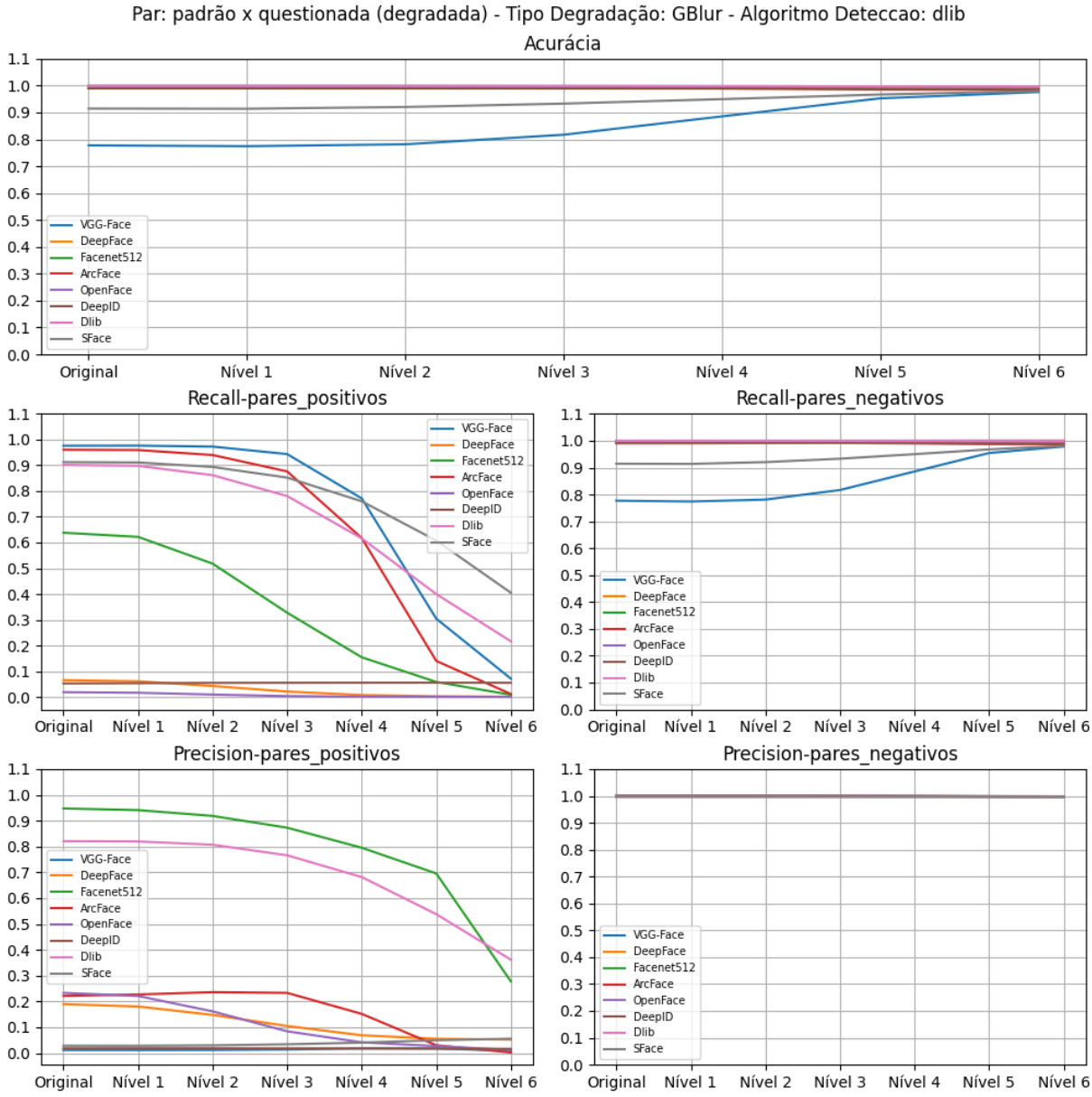


Figura 5.4: Degradação: Borramento Gaussiano - Reconhecimento: todos - Detecção: dlib.

Outro ponto que vale destacar é que foi possível observar diferentes comportamentos para os pares de imagens gerados, melhor descrito conforme a seguir:

1. Par 1: Imagem Padrão (não degradada) x Cópia da Imagem Padrão (degradada)

Ao examinar a geração do Par 1, composto por uma imagem padrão (não degradada) em relação a uma cópia da imagem padrão (degradada), o estudo mostrou-se relevante, pois quantificou o declínio no desempenho do sistema para cada tipo e intensidade de degradação. Como duas imagens idênticas formam o par, uma não degradada e outra degradada, o par é formado apenas pela mesma pessoa. Então, por conta disso, não há informações referentes a pares negativos neste item. Quanto ao impacto das degradações nos pipelines, observou-se que, como esperado, as degradações mistas tiveram um impacto maior em comparação às degradações simples. As combinações de degradações que exibiram o maior nível (Figure 5.5) de impacto nos pipelines, respectivamente, durante os experimentos, foram: GBlur →Dark →Down →JPEG; GBlur →Dark →JPEG; GBlur →Bright →JPEG; GBlur →Dark →Down.

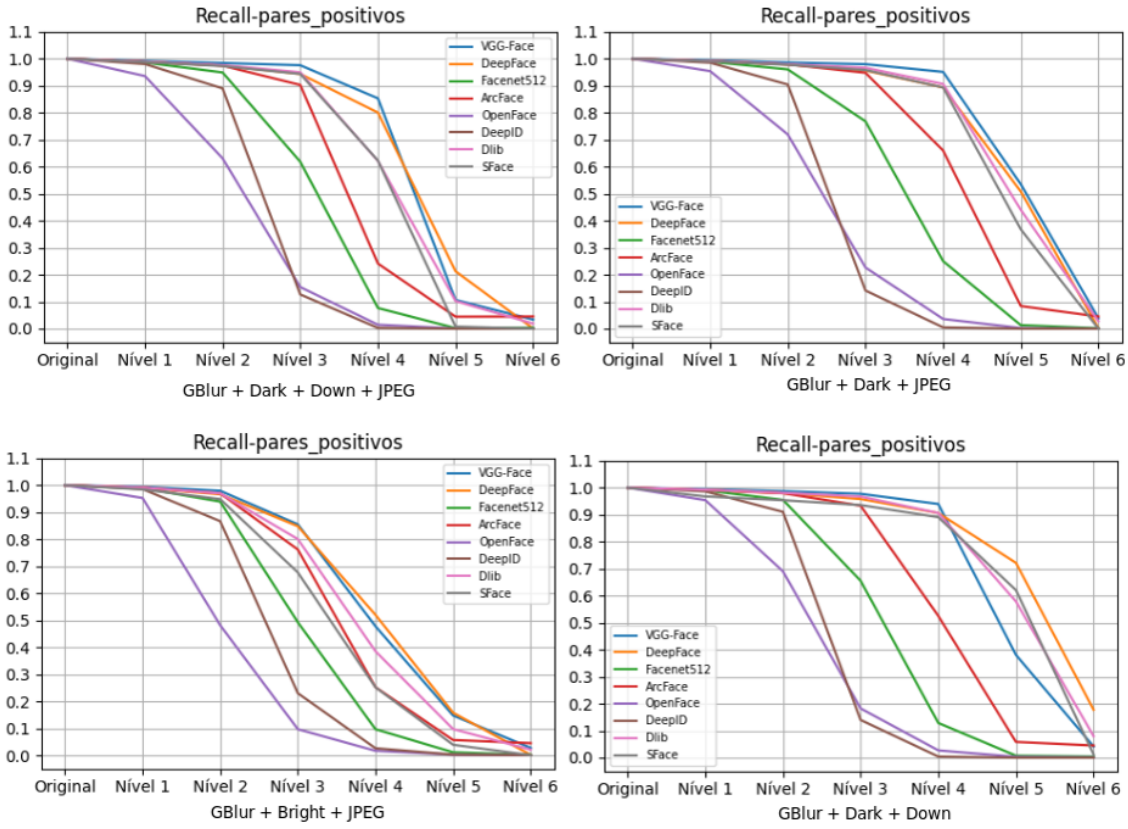


Figura 5.5: Maiores impactos para o Par 1 (todos os modelos) (mtcnn).

Por outro lado, as degradações que tiveram os menores impactos no experimento, exibidas na Figura 5.6, foram JPEG, Dark, and Bright.

2. Par 2: Imagem Padrão (sem degradação) x Imagem Questionada (com degradação)

   O estudo analisou minuciosamente o Par 2, composto por uma imagem padrão (não degradada) comparada a uma imagem questionada (degradada). Este exame

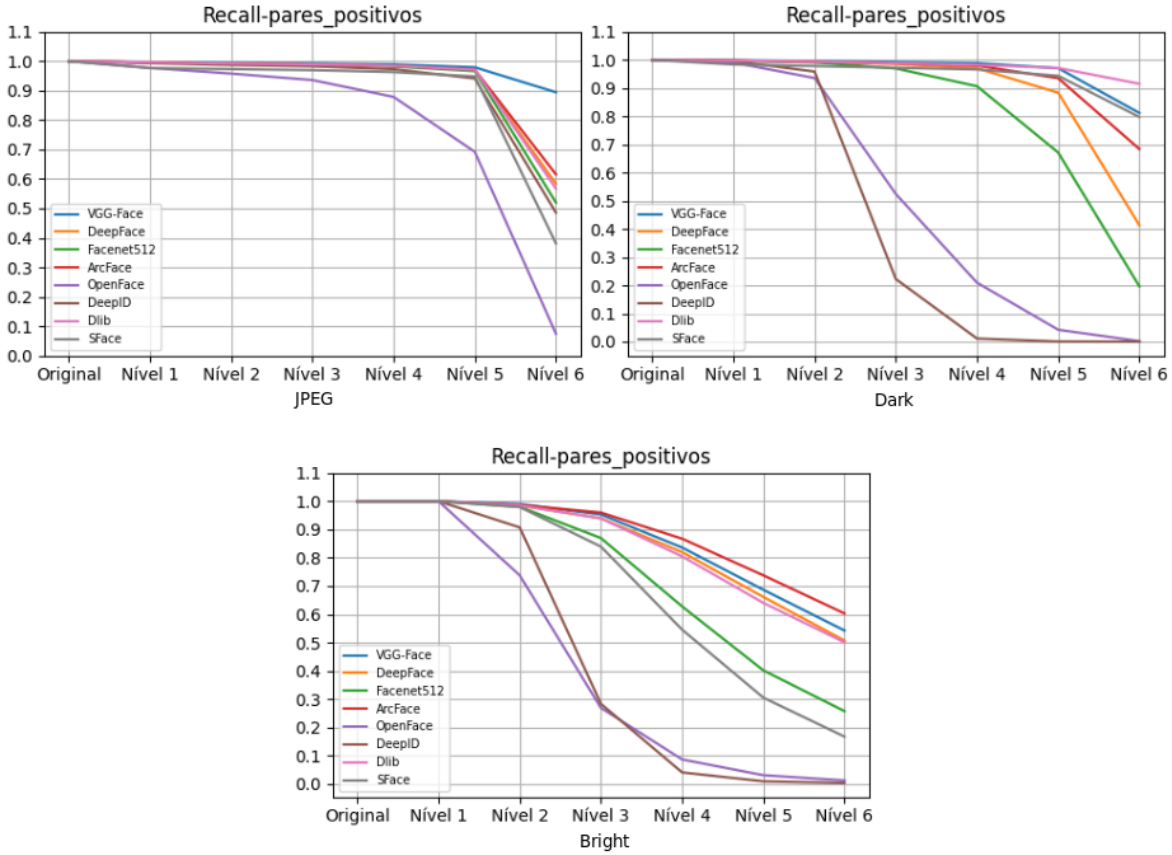


Figura 5.6: Menores impactos para o Par 1 (todos os modelos) (mtcnn).

revelou-se altamente pertinente, pois além de medir com precisão a diminuição no desempenho do sistema associada a cada tipo específico e intensidade de degradação, caracteriza-se por ser o cenário mais frequentemente encontrado nos exames periciais, aumentando a relevância do estudo. Em termos de degradação de impacto, os resultados da pesquisa indicaram quais produziram mais impacto nos pipelines, respectivamente (Figura 5.7): GBlur →Dark →Down →JPEG; GBlur →Dark →JPEG; GBlur →Bright →JPEG; GBlur →Dark →Down.

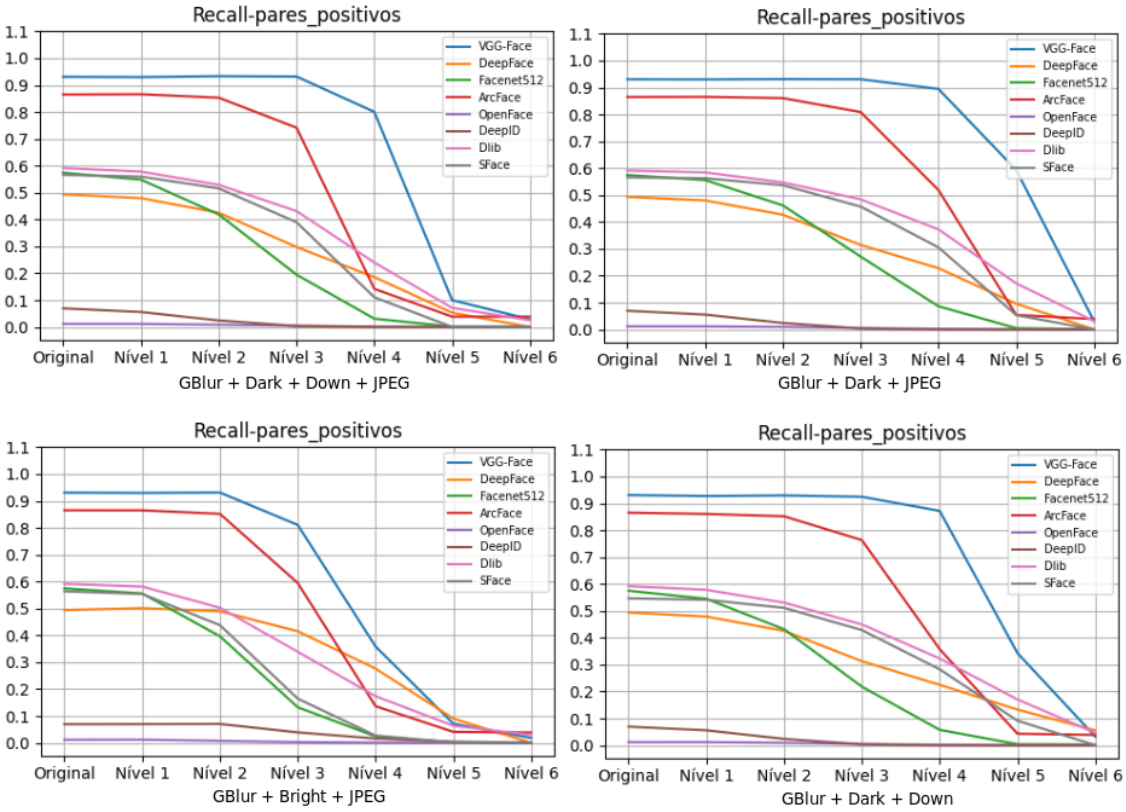


Figura 5.7: Maiores impactos para o Par 2 (todos os modelos) (mtcnn).

As degradações que tiveram menos impacto foram (Figure 5.8): JPEG, Dark, and Bright.

3. Par 3: Imagem Padrão (com degradação) x Imagem Questionada (com degradação)

   No que diz respeito à geração do terceiro par de imagens, que inclui uma imagem padrão degradada em relação a uma imagem questionada degradada, a avaliação não apenas quantificou o impacto das degradações nos sistemas de reconhecimento facial, mas também revelou um comportamento alarmante dos algoritmos. Observou-se que, em determinado momento, os algoritmos identificaram todas as imagens como indistinguíveis, independentemente das pessoas comparadas. Isso ocorreu quando a intensidade da degradação atingiu um nível severo, fazendo com que os algoritmos

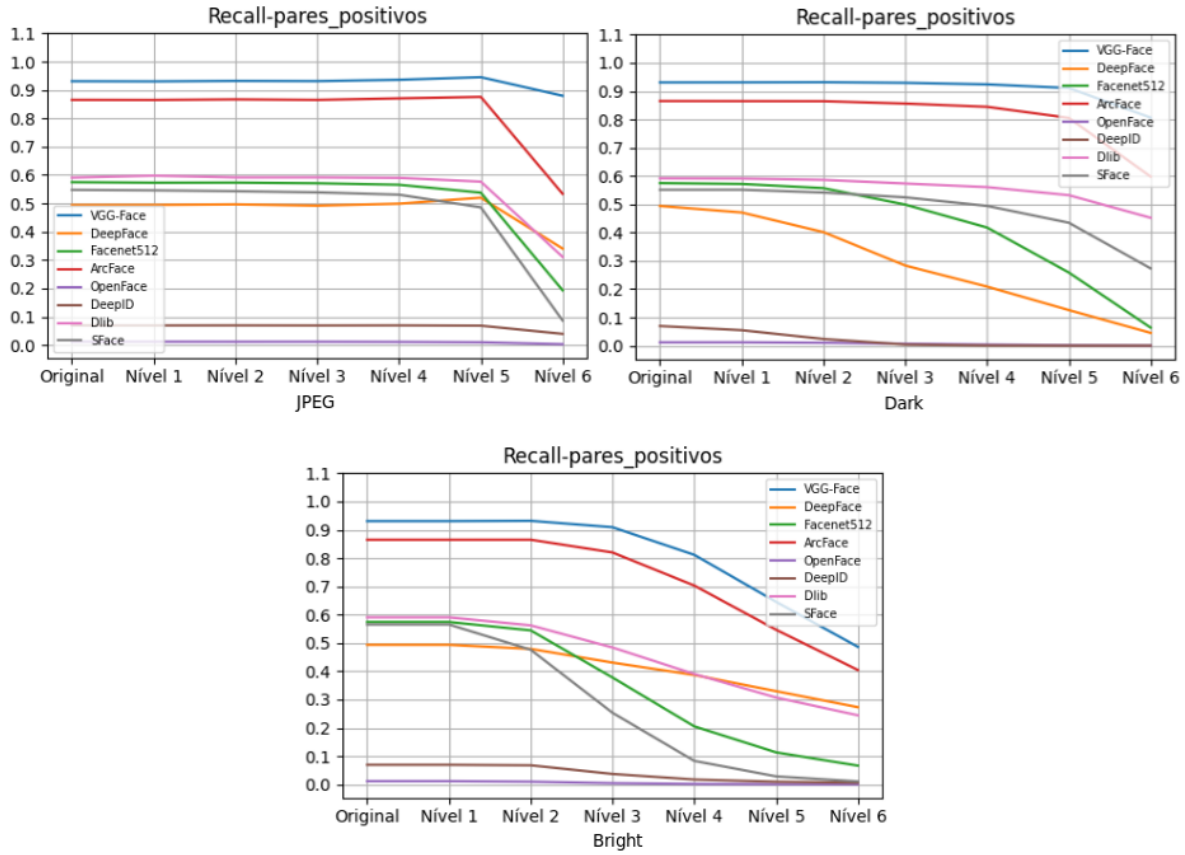


Figura 5.8: Menores impactos para o Par 2 (todos os modelos) (mtcnn).

perdessem a capacidade de diferenciar entre indivíduos diferentes e, em vez disso, os trataram como a "mesma pessoa". Esse comportamento levanta preocupações graves, especialmente em aplicações que operam em ambientes não controlados, já que ambas as imagens podem ter sofrido degradação antes de serem processadas pelo algoritmo de reconhecimento facial. Todas as observações podem ser vistas na Figura 5.9.

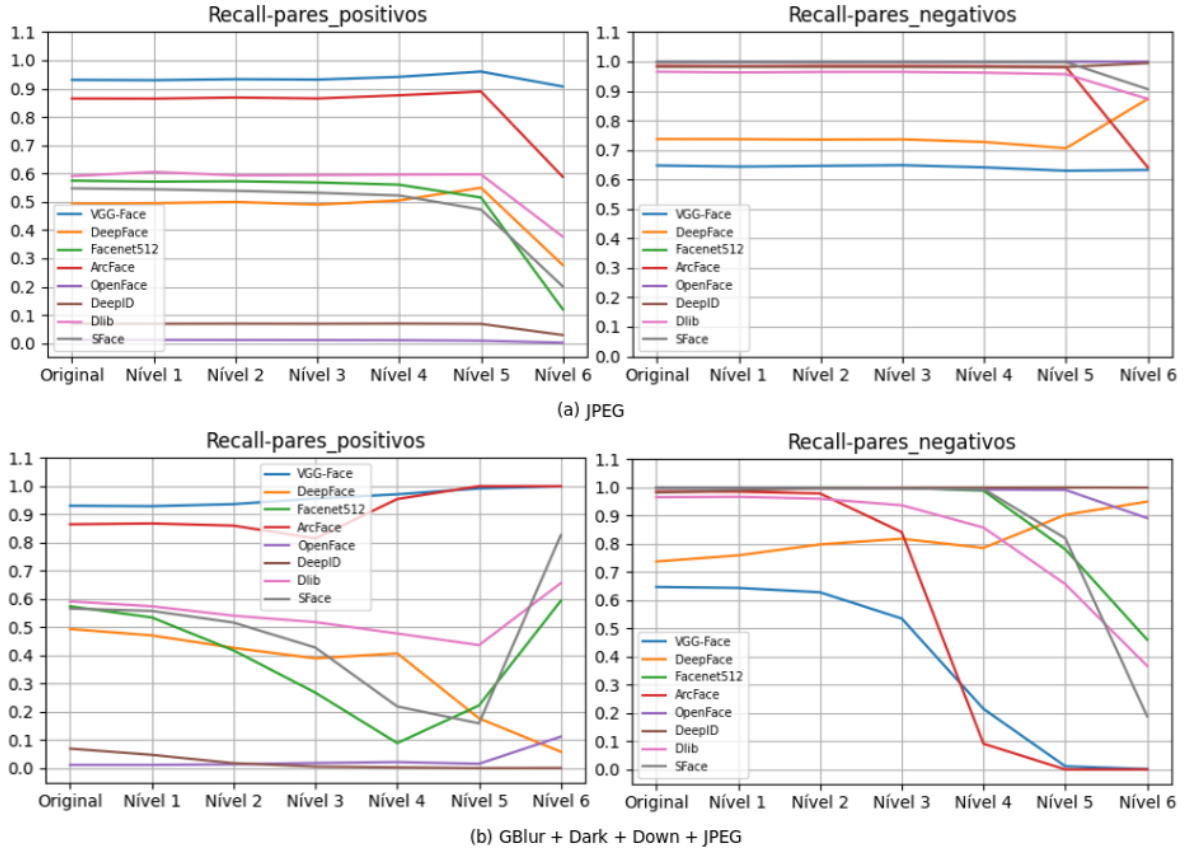


Figura 5.9: Menores (a) e maiores (b) impactos no Par 3 (todos os modelos) (mtcnn).

Uma análise dos gráficos de desempenho dos algoritmos revela um padrão consistente e preocupante: Para os algoritmos que inicialmente demonstram alto desempenho em imagens não degradadas, observa-se uma diminuição gradual na pontuação métrica à medida que a intensidade da degradação aumenta, o que está de acordo com as expectativas. No entanto, em um limiar específico, como evidenciado no gráfico (terceira imagem do gráfico 5.9) que representa amostras de pares positivos, a pontuação em declínio sobe inesperadamente para 100%. Isso indica que todos os pares positivos (representando a mesma pessoa) são classificados corretamente como idênticos. Esse comportamento levanta suspeitas, pois contradiz a tendência esperada de que imagens degradadas resultem em desempenho reduzido do algoritmo.

Ao mesmo tempo, uma análise do gráfico que representa amostras de pares negativos demonstra que, no mesmo limiar em que a curva de pares positivos começa a subir, a curva de pares negativos (quarta imagem do gráfico 5.9) cai para 0, indicando que nenhum par negativo foi identificado como "diferentes". Quando a curva de pares positivos atinge 1 e a curva de pares negativos atinge 0, os algoritmos perdem a capacidade de identificar corretamente pares como um todo, categorizando efetivamente todos os pares como a "mesma pessoa".

O gráfico exibido na Figura I.24, o qual mostra o algoritmo VGGFace para todas as degradações deste estudo, pode ser analisado em conjunto para melhor entendimento deste comportamento supramencionado.

Uma tendência semelhante é observada para algoritmos com desempenho inicialmente baixo. No entanto, como esses algoritmos já apresentam baixo desempenho em imagens não degradadas, o declínio inicial esperado no desempenho é menos perceptível. Somente após ultrapassar um limiar específico (nível de intensidade) é que a elevação na curva de pares positivos e a queda na curva de pares negativos se tornam evidentes.

Essas observações destacam limitações significativas e levantam preocupações quanto à confiabilidade dos algoritmos e à capacidade de identificar e distinguir com precisão entre pares, especialmente em condições de imagem altamente degradada.

Em um cenário hipotético, um especialista que analisa os resultados poderia erroneamente concluir que duas imagens pertencem à mesma pessoa. No entanto, como demonstrado nos experimentos, o sistema poderia já ter perdido a capacidade de diferenciar entre indivíduos. No contexto criminal, esse comportamento pode levar o especialista a apoiar suspeitas ou, em casos mais graves, resultar em prisões injustas. Ao analisar imagens degradadas, eles podem considerar a imagem de um suspeito e a de um cidadão inocente como pertencentes à mesma pessoa, iniciando investigações, processos legais e até mesmo condenações com base em informações interpretadas de forma errônea.

Perspectiva do Modelo

Analisando os resultados da perspectiva dos modelos, é possível observar diferentes comportamentos. A análise foi separada por base de dados, conforme a seguir.

1. LFW Dataset

   O dataset LFW consiste em imagens capturadas em um ambiente não controlado, tornando mais desafiador para os modelos fornecer correspondências precisas. Para

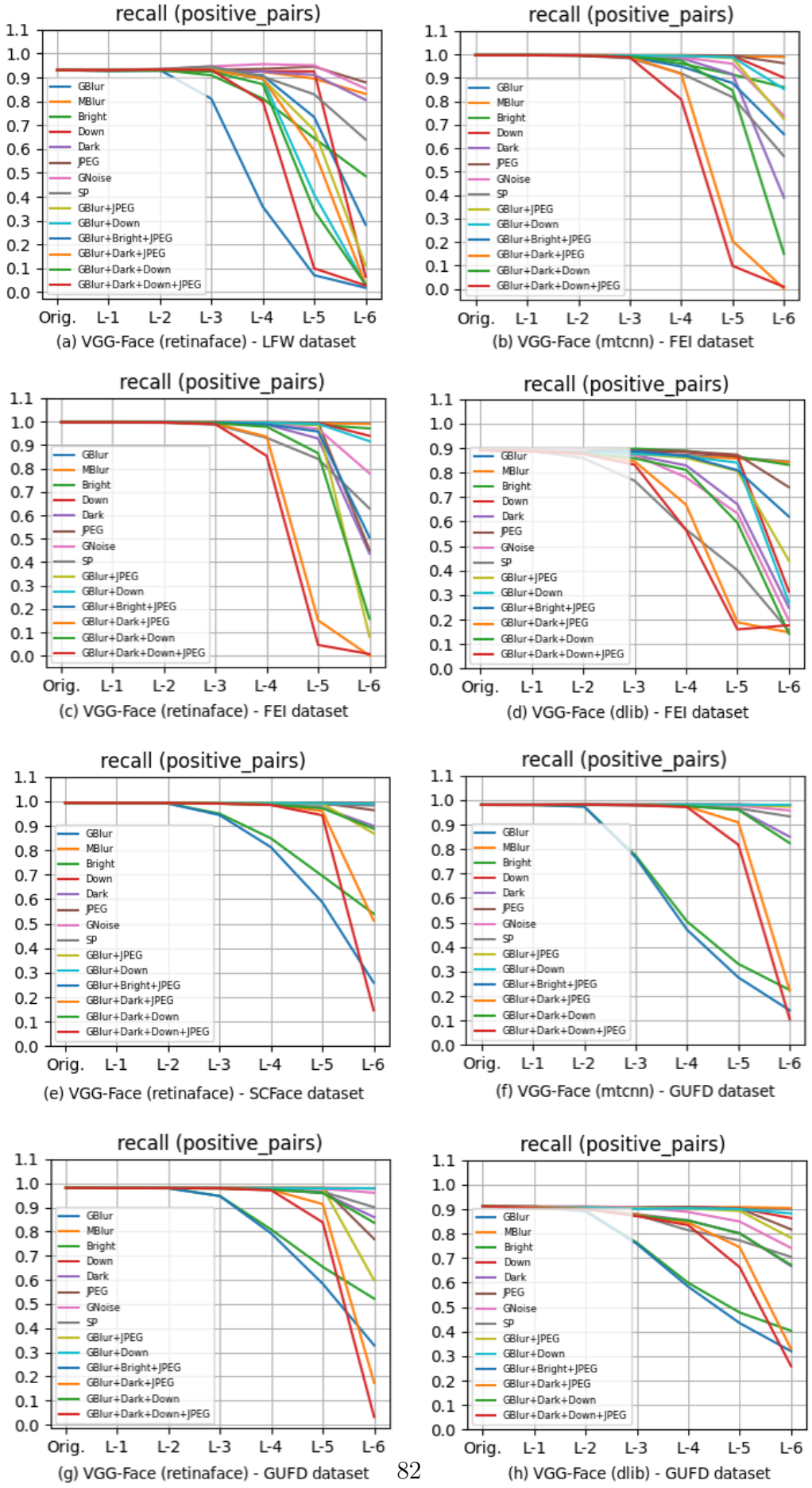


Figura 5.10: Impacto das degradações para pipelines específicos - Par 2

a etapa inicial de imagens não degradadas, os pipelines em ordem de melhor desempenho (com uma taxa de sucesso próxima a 100%) foram: VGGFace+(todos os 3 detectores) (mtcnn: 97,89%, dlib: 97,61% e retinaface: 93,07%), ArcFace+dlib (96,05%), ArcFace+mtcnn (92,62%), SFace+dlib (91,25%) e Dlib+dlib (90,14%). Para a última etapa, com o nível mais alto de degradação, os melhores pipelines (com menos curvas de degradação a 0%) foram: VGGFace+(todos os 3 detectores) (retinaface: 0 curvas, mtcnn: 5 curvas e dlib: 5 curvas) e SFace+dlib (7 curvas). Neste ponto, é digno de nota que o VGGFace alcançou os melhores resultados entre todos os modelos de reconhecimento facial veja a Figura 5.10-(a).

2. FEI Dataset

   O dataset FEI consiste em imagens capturadas em um ambiente controlado. Para a etapa inicial de imagens degradadas, os pipelines com melhor desempenho foram, respectivamente: VGGFace+retinaface (99,68%), VGGFace+mtcnn (99,63%), ArcFace+retinaface (95,06%), ArcFace+mtcnn (94,09%), Dlib+dlib (92,36%), Facenet512+retinaface (88,96%) e Facenet512+mtcnn (87,06%). É notável que VGGFace+mtcnn e VGGFace+retinaface alcançaram os melhores resultados entre todos os modelos de reconhecimento facial (veja a Figura 5.10-(b) e (c)). Na última etapa de imagens degradadas, os melhores pipelines foram VGGFace+(todos os 3 detectores) (dlib: 0 curvas, mtcnn: 2 curvas e retinaface: 2 curvas)). Nesse ponto, VGGFace+dlib obteve os melhores resultados entre todos os modelos de reconhecimento facial (veja a Figura 5.10-(d)).

3. SCFace Dataset

   Para a etapa inicial de imagens degradadas, os pipelines com melhor desempenho foram: VGGFace+retinaface (99,43%) e VGGFace+mtcnn (98,70%). Nesse caso, VGGFace+retinaface obteve os melhores resultados (veja a Figura 5.10-(e)). Para a última etapa de imagens degradadas, os melhores pipelines foram: Arcface+(todos os 3 detectores)(0 curvas) e VGGFace+(todos os 3 detectores)(0 curvas). Novamente, VGGFace+retinaface foi o melhor modelo.

4. GUFD Dataset

   Para a etapa inicial de imagens degradadas, os pipelines com melhor desempenho foram: VGGFace+(todos os 3 detectores) (mtcnn: 98,10%, retinaface: 98,08%, dlib: 91,17%) e Dlib+dlib (94,06%). Neste caso, VGGFace+mtcnn e VGGFace+retinaface alcançaram os melhores resultados (veja a Figura 5.10-(f) e (g)). Na última etapa de imagens degradadas, os melhores pipelines foram: Arcface+(todos os 3 detectores)(0 curvas), VGGFace+dlib (0 curvas) e VGGFace+mtcnn (0 curvas). E VGG-

Face+dlib obteve os melhores resultados entre todos os modelos de reconhecimento facial (veja a Figura 5.10-(h)).

Como exposto, com base nos resultados experimentais de todos os conjuntos de imagens testados, foi consistentemente observado que o VGGFace apresentou o melhor desempenho em imagens não degradadas ou em níveis mais baixos de degradação. Além disso, o VGGFace demonstrou uma superior robustez quando confrontado com os níveis mais intensos de degradação de imagem.

Quantificando a afirmação do parágrafo anterior, o VGG-Face obteve uma acurácia média de 94% no nível 3, considerando todas as degradações, enquanto o segundo algoritmo melhor colocado, obteve 81%. Ainda, o VGG-Face obteve acurácia média de 34% para no nível 6, considerando todas as degradações, enquanto o segundo algoritmo melhor colocado, obteve 16%.

5.2.3 Gráficos de Impacto dos Modelos de Detecção Facial

Aqui exibimos os impactos referentes aos modelos de detecção facial, para o *dataset* considerado mais difícil, LFW, conforme Figuras I.103 e I.104.

5.2.4 Análises dos Algoritmos de Detecção Facial

A análise dos algoritmos foi realizada comparando os 3 (três) algoritmos de detecção facial, no tocante à detecção das faces, ou seja, analisamos a acurácia do algoritmo à medida que as degradações se intensificavam.

Conforme explicado, capturamos o momento da quebra do pipeline do sistema de reconhecimento facial, ou seja, o momento em que a degradação é tão intensa que o algoritmo de detecção facial não é mais capaz de detectar a face da imagem. Desta forma, os gráficos exibem a curva de acurácia, demonstrando a robustez de cada algoritmo frente a cada degradação.

Através dos gráficos, observou-se que o algoritmo dlib é menos robusto em comparação com mtcnn e retinaface, obtendo o pior desempenho em 9 (nove) das 14 (quatorze) degradações estudadas.

Ainda, observou-se que determinados algoritmos são mais robustos em certas degradações, por exemplo, o mtcnn é mais robusto frente à Compressão JPEG, Sal e Pimenta. Ainda nessa linha, o algoritmo ainda se mostrou o mais robusto na maioria nas degradações em sequência que incluíam a Compressão JPEG: Ruído Gaussiano → Compressão JPEG, Ruído Gaussiano → Escurecimento → Compressão JPEG e Ruído Gaussiano → Escurecimento → Redução de Tamanho → Compressão JPEG.

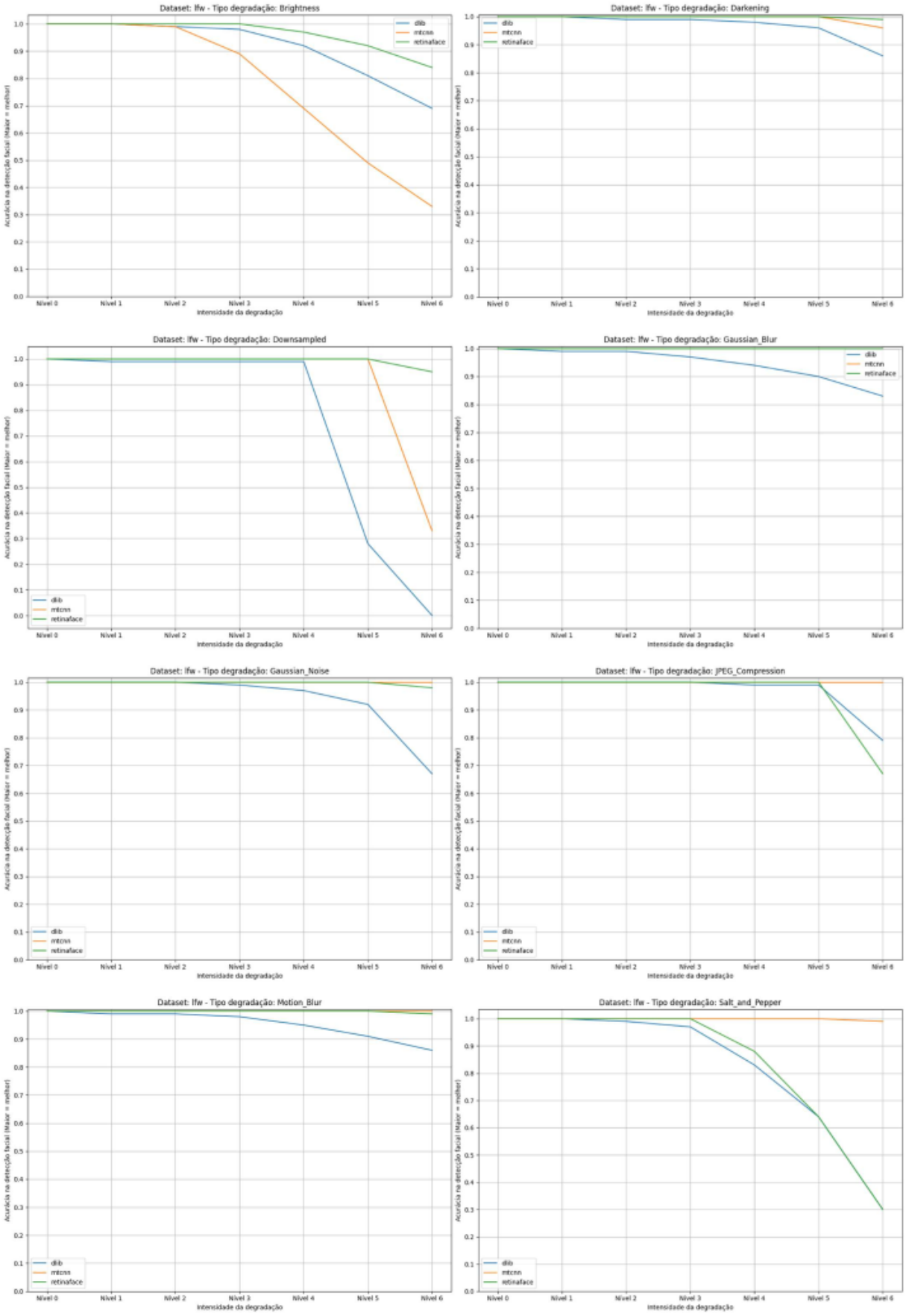


Figura 5.11: Desempenho dos algoritmos de detecção facial (degradações simples)

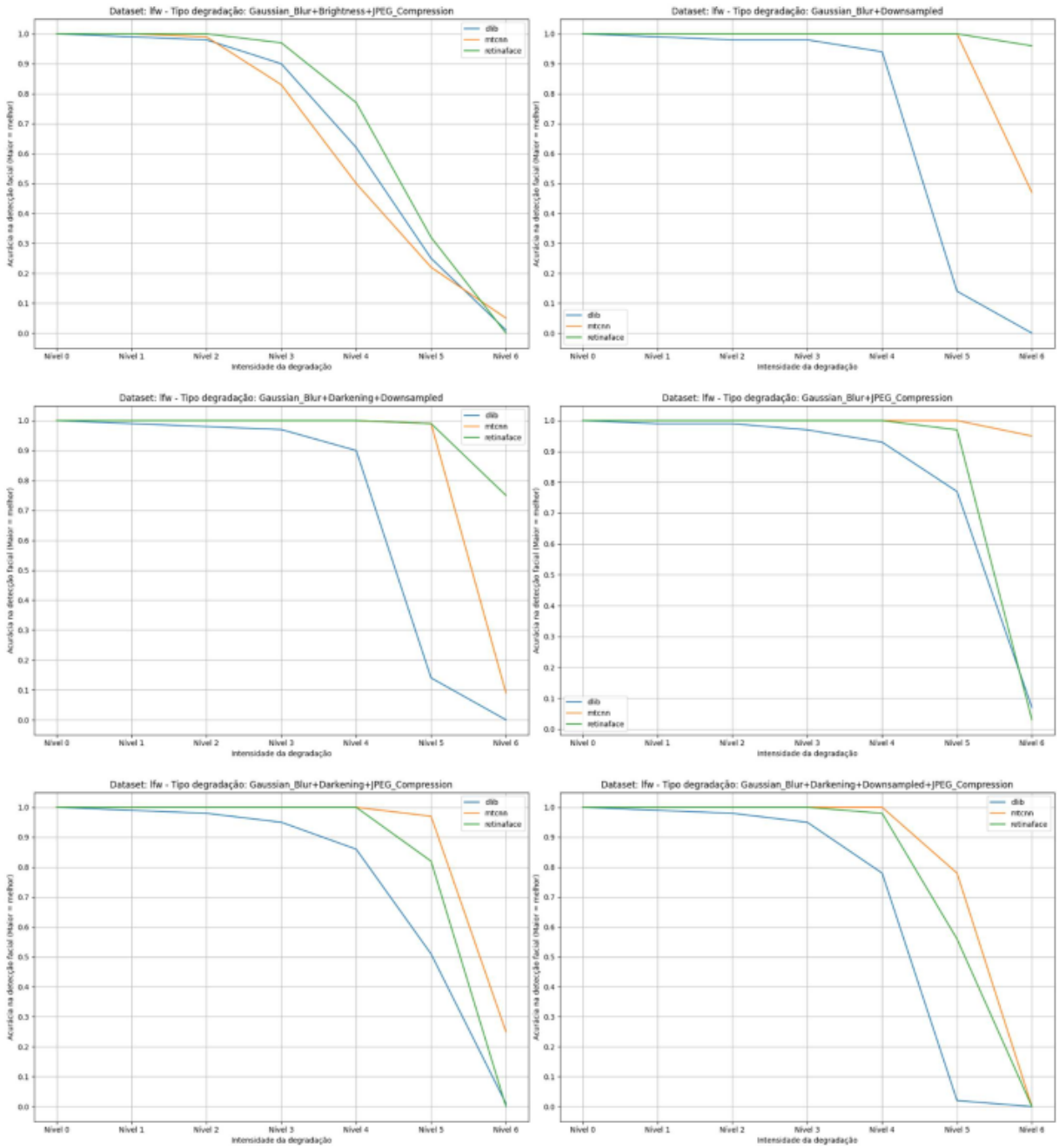


Figura 5.12: Desempenho dos algoritmos de detecção facial (degradações em sequência)

Já o algoritmo retinaface se provou mais robusto frente às degradações de Brilho e Redução de Tamanho que o mtcnn. Nas degradações em sequência, é preciso observar cada caso, visto que a intensidade de cada tipo de degradação pode prejudicar e/ou favorecer uso de determinado algoritmo de detecção facial. Como por exemplo na degradação Ruído Gaussino → Brilho → Compressão JPEG, onde o retinaface foi superior ao mtcnn.

Em regra, nas degradações que incluíam Compressão JPEG, o algoritmo mtcnn foi superior ao retinaface, entretanto observam-se exceções como na Ruído Gaussino → Brilho → Compressão JPEG, provavelmente devido à incidência do Briho, onde o retinaface apresentou mais robustez.

Desta forma, fica evidente que a depender das características das imagens em que a aplicação de reconhecimento facial estiver executando, a escolha do algoritmo de detecção facial tem influência direta no resultado final do sistema.

5.3 Resultados da Fase 2 - Definição de Modelos para Detecção de Degradações

Nesta seção exibiremos os resultados da Fase 2 do projeto, ou seja, o resultado final dos modelos de aprendizado profundo treinados na tarefa de identificação do tipo de degradação e intensidade eventualmente presentes na imagem da face.

A execução dos experimentos possibilitou o treinamento de 20 (vinte) modelos CNN para a tarefa de identificação de degradação em imagens. As 10 (dez)arquiteturas empregadas passaram por treinamento tanto *from scratch* quanto por *transfer learning*. Conforme mencionado, 4 (quatro) GPUs Tesla V100 foram utilizadas para treinamento de modelos, permitindo a análise dos tempos de treinamento de cada modelo.

Após a montagem do *dataset* de teste e treinamento, um total de 20 (vinte) modelos foi treinado. Cada modelo foi treinado por 100 épocas, com os mesmos hiperparâmetros (conforme Tabela 5.1), com a mesma base de dados, descrita na subseção anterior. Além disso, os modelos foram treinados em placas de vídeo idênticas (foram disponibilizadas 4 placas de vídeo Nvidia Tesla V100 para este experimento), instaladas em 2 servidores clones (foram disponibilizados 2 servidores idênticos). Desta forma, espera-se que o ambiente de execução das rotinas tenha sido o mais próximo possível para permitir comparações entre os treinamentos dos modelos.

Para sintetizar esta fase da proposta, conforme mencionado, o treinamento dos 20 modelos foi dividido entre 2 servidores idênticos, cada um contendo 2 placas gráficas idênticas. Ainda, todos os modelos foram treinados com os mesmos hiperparâmetros, pela mesma quantidade de épocas e com mesmos dados.

Os resultados alcançados são apresentados na Tabela 5.2.

Hiperparâmetro	**Valor**
Taxa de aprendizagem (LR)	0.001
Função de erro	CrossEntropyLoss
Algoritmo de otimização	Adam
Weight Decay	-
Momentum	(0.95, 0.85, 0.95)

Tabela 5.1: Hiperparâmetros utilizados

Tabela 5.2: Métricas de Treinamento e Teste.

Modelos	**From Scratch (FS)**				**Transfer Learning (TL)**			
	Treinamento			*Teste*	*Treinamento*			*Teste*
	Erro	***Acurácia***	***Tempo***	***Acurácia***	***Erro***	***Acurácia***	***Tempo***	***Acurácia***
ResNet-18	0.094343	0.953567	28h38m	**0.710109**	0.262642	0.887048	22h20m	0.900774
ResNet-50	0.091517	0.953864	86h20m	0.900762	0.120287	0.941282	68h13m	0.937476
ResNet-152	0.079546	0.960381	204h53m	0.887542	0.09232	0.953169	162h58m	**0.941531**
DenseNet-121	0.099505	0.952477	81h01m	0.905889	0.138116	0.93606	63h38m	0.931579
DenseNet-201	0.090869	0.953982	125h55m	0.924876	0.094926	0.951131	104h03m	**0.94279**
VGG-19	0.119625	0.950305	148h16m	0.920368	0.189371	0.912851	110h38m	0.923296
Inception-v3	0.094049	0.953213	56h33m	**0.941602**	0.202204	0.910502	41h31m	0.913517
Inception-v4	0.097519	0.953577	118h13m	0.926156	0.153622	0.930508	82h36m	0.923035
Xception-71	0.100883	0.95185	284h23m	0.939818	0.103266	0.950655	205h45m	0.939299
MobileNet-v2-100	0.115711	0.946205	40h	0.933991	0.180818	0.917305	26h55m	0.9183

Para melhor visualização e análise, os resultados de acurácia e tempo de treinamento foram plotados em um gráfico, conforme exibido na Figura 5.13.

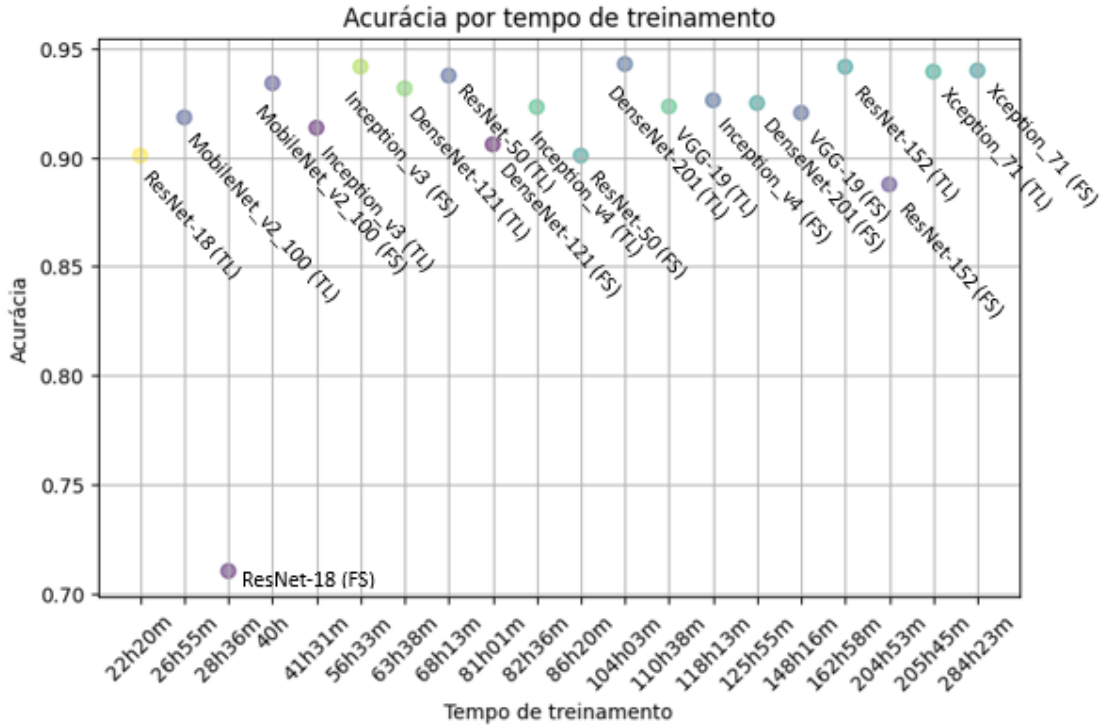


Figura 5.13: Acurácia x Tempo de treinamento para 100 épocas.

Ainda, a fim de facilitar a análise geral dos modelos treinados, os modelos foram classificados conforme os resultados dos obtidos no conjunto de dados de teste, no tocante à métrica acurácia, da seguinte forma: Acurácia até 80%; Acurácia entre 80% e 90%; Acurácia entre 90% e 94%; e Acurácia acima de 94%.

5.3.1 Acurácia até 80%

Neste grupo, apenas o modelo ResNet-18 (FS) obteve precisão abaixo de 80% (atingindo 71%). Provavelmente devido à profundidade das camadas, a rede não conseguiu aprender os parâmetros de forma a separar as classes de forma eficiente. Vale ressaltar que a mesma arquitetura, quando treinada utilizando *transfer learning*, atingiu 90% de precisão na classificação das imagens degradadas.

5.3.2 Acurácia entre 80% e 90%

Neste caso, apenas o modelo ResNet-152 (FS) está presente, alcançando uma precisão de 88% ao final do treinamento. Surpreendentemente, o modelo ResNet com 152 camadas

(FS), que é o mais profundo da família ResNet, não alcançou as melhores posições entre os resultados, sendo superado, por exemplo, pelo modelo ResNet-50 (FS), que é da mesma família e tem menos camadas. Além disso, a versão treinada por transferência de aprendizagem do modelo ResNet-152 também ficou entre as 3 primeiras neste experimento. Neste ponto, mais estudos poderiam ser realizados para entender melhor por que a abordagem *from scratch* não alcançou uma precisão maior do que a obtida.

5.3.3 Acurácia entre 90% e 94%

Este grupo engloba a maioria dos modelos treinados, pois a maioria deles alcançou acurácia entre 90% e 94%. Os modelos neste grupo incluem ResNet-18 (TL), ResNet-50 (FS e TL), DenseNet (FS e TL), DenseNet-201 (FS), VGG-19 (FS e TL), Inception-V3 (TL), Inception-V4 (FS e TL), Xception-71 (FS e TL) e MobileNet-V2 (FS e TL).

5.3.4 Acurácia acima de 94%

Aqui estão os três principais modelos do experimento: ResNet-152 (TL), DenseNet-201 (TL) e Inception-V3 (FS). Todos os modelos obtiveram precisão acima de 94%, sendo que o modelo DenseNet-201 (TL) atingiu uma precisão de 94,27%, superando ligeiramente os demais (Inception-V3 (FS): 94,16% e ResNet-152 (TL): 94,15 %).

Para compreender melhor a convergência desses 3 modelos, é importante analisar a curva de aprendizado de cada modelo durante o treinamento. Assim, o gráfico abaixo representa as curvas de erro e precisão ao longo de 100 épocas (Figura 5.16):

5.4 Discussões dos Resultados

Nesta seção explanaremos alguns pontos que carecem de maior discussão a respeito dos resultados obtidos.

5.4.1 Fase 1 - Avaliação da Influência das Degradações em Modelos Profundos

O trabalho avaliou 24 pipelines diferentes de reconhecimento facial frente à 14 tipos de degradações diferentes. Durante o experimento, foram coletadas 3 métricas: *accuracy*, *precision* e revocação. No decorrer das análises, verificou-se que a métrica *accuracy* seguia a tendência dos pares negativos, tendo em vista que a quantidade de pares negativos e muito superior à quantidade de pares positivos. Os conjuntos de dados de faces possuem algumas imagens por pessoa, de modo que a quantidade de pares positivos é muito menor

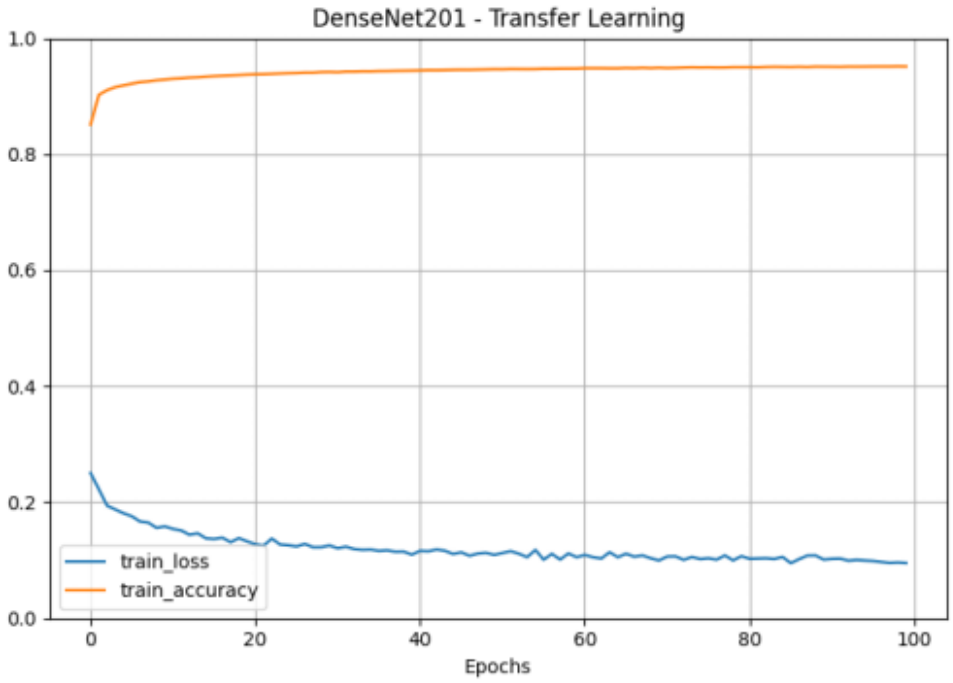


Figura 5.14: Curva de treinamento do modelo DenseNet-201 (TL).

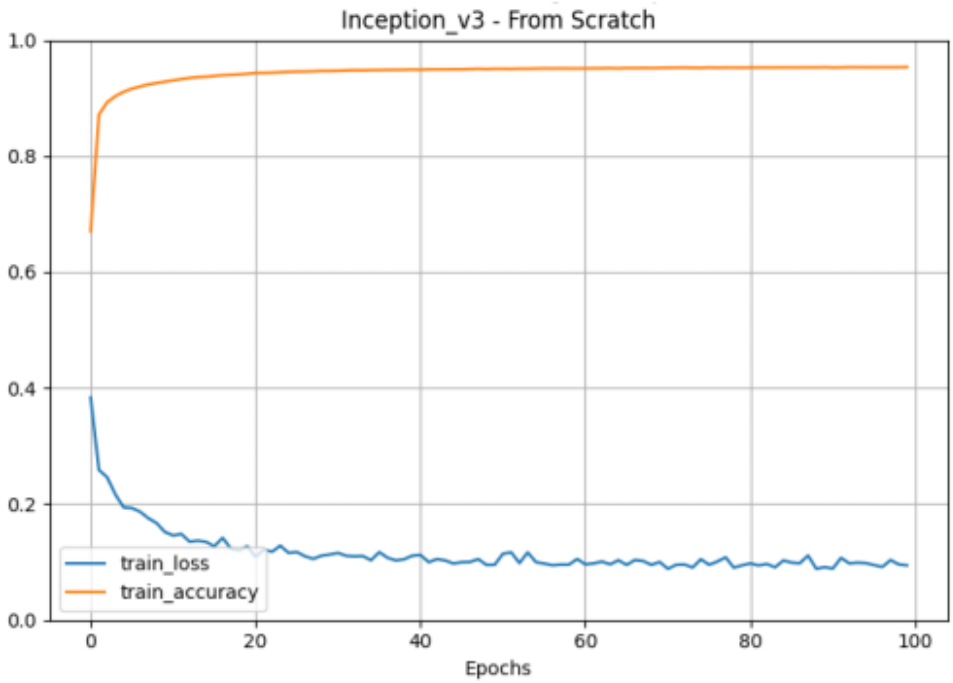


Figura 5.15: Curva de treinamento do modelo Inception-v3 (FS).

do que de pares negativos. Como a métrica da acurácia leva em conta todos os pares, o resultado final da métrica segue o resultado dos pares negativos. Por este motivo, a maior parte das análises foi realizada em cima da métrica revocação.

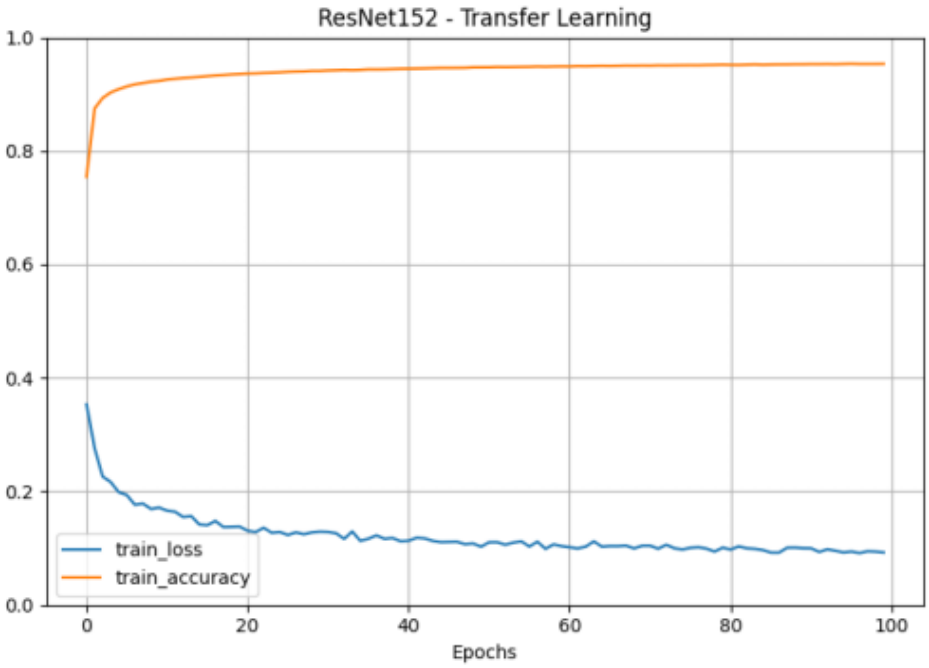


Figura 5.16: Curva de treinamento do modelo ResNet-152 (TL).

O experimento avaliou o impacto das degradações nos sistemas de reconhecimento facial, e, de maneira geral, pode-se observar uma queda no desempenho, conforme esperado. Para os Pares 1 e 2, o objetivo do experimento se cumpriu ao analisar e quantificar o impacto no desempenho, conforme descrito ao longo do trabalho. Entretanto, para o Par 3, que envolve imagens padrão e questionadas degradadas, tivemos um achado importante. Observou-se uma tendência que possivelmente causa erros de interpretação se analisada de forma isolada, sem outras observações em conjunto.

É o caso quando a intensidade da degradação é tão intensa (a partir do nível 3 de intensidade) que ocasiona ao algoritmo inferir que duas imagens pertencem a mesma pessoa, mesmo que sejam de pessoas diferentes. Ou seja, a (mesma) degradação nas imagens é tão intensa, que o algoritmo encontra similaridade entre as imagens, mesmo quando uma face mal pode ser percebida. Vale reforçar que este padrão foi identificado para diversos pipelines de reconhecimento facial estudados.

Outro ponto, quantificando o desempenho dos algoritmos, sobre especificamente o conjunto de dados LFW, o qual foi considerado o mais desafiador neste estudo, verificamos que os algoritmos que tiveram melhor desempenho no nível 3 de intensidade foram: VGG-Face (94% de acurácia), ArcFace (81%) e Dlib (45%). Já para o nível mais intenso de degradação, nível 6, os 3 melhores algoritmos foram: VGG-Face (34% de acurácia), Dlib (16%) e ArcFace (13%). Estes números levam em conta o desempenho nas 14 (quatorze) degradações estudadas.

Quando observamos sob o ponto de vista dos tipos de degradações, as 3 (três) degradações que apresentaram mais impacto no desempenho dos pipelines foram as degradações em sequência: Borramento Gaussiano → Escurecimento → Redução de Tamanho → Compressão JPEG, Borramento Gaussiano → Brilho → Compressão JPEG e Borramento Gaussiano → Escurecimento → Redução de Tamanho. E as 3 (três) que tiveram menor impacto foram as degradações simples: Compressão JPEG, Escurecimento e Brilho.

Em relação aos algoritmos de detecção facial, verificamos que MTCNN foi o que teve melhor desempenho, seguido do retinaface. Das 14 (quatorze) degradações analisadas, o algoritmo dlib ficou em último lugar em 9 (nove) delas, demonstrando assim ser o algoritmo de detecção facial com maior sensibilidade às degradações.

Como pontos positivos desta fase podemos destacar a quantificação do desempenho dos algoritmos de reconhecimento facial em relação às degradações frequentemente vistas nas imagens. Além disso, após este estudo, é possível medir o impacto, e eventualmente, concluir que determinado algoritmo não deve ser utilizado quando as degradações presentes no material impactam significativamente o resultado.

De forma negativa, entretanto, entende-se que este estudo tem escopo específico, e em caso de novos algoritmos e/ou degradações diferentes das aqui estudadas, novos experimentos devem ser realizados para verificar as curvas de impacto.

Outro ponto negativo do estudo se dá pela necessidade de *hardware* específico e tempo de processamento para calcular as curvas de impacto. Exceto pelos servidores disponibilizados pela Universidade de Brasília, dificilmente a pesquisa seria viável em uma infraestrutura mais modesta.

5.4.2 Fase 2 - Definição de Modelos para Detecção de Degradações

Os resultados dos treinamentos realizados tanto na abordagem *from scratch* quanto com *transfer learning* demonstraram que na abordagem com *transfer learning*, o treinamento é realizado de forma mais rápida, consumindo menos tempo de processamento. Entretanto, essa velocidade no treinamento não significou, necessariamente, melhor desempenho, visto que nas 10 (dez) arquiteturas estudadas, a abordagem com *transfer learning* obteve melhor acurácia em apenas 6 (seis) delas.

Em relação aos valores finais de acurácia, os 3 (três) modelos que obtiveram melhor resultado foram DenseNet-201 (TL), Inception-V3 (FS) e Resnet-152 (TL). Aqui observamos que o modelo Inception-v3 obteve curvas de aprendizado com mais picos que os modelos DenseNet-201 e Resnet-152. Isso possivelmente se dá pela inicialização dos pe-

sos de cada modelo, visto que DenseNet-201 e Resnet-152 foram treinados com *transfer learning*.

De forma positiva, os modelos de degradação podem automatizar a detecção da degradação presente na imagem, e em conjunto com os resultados obtidos na fase anterior, estimar o impacto que eventuais degradações incidem em determinado algoritmo. Assim, de posse da informação do impacto estimado, cabe ao especialista decidir pelo uso de determinado algoritmo, trocar de algoritmo ou realizar apenas uma análise subjetiva no material recebido.

Como ponto negativo, da mesma forma que o estudo da Fase 1, para qualquer nova degradação, o modelo deve ser retreinado para ajustar os pesos, permitindo assim a detecção. Neste tocante, são cabíveis as mesmas considerações a respeito da especificidade de *hardware* e tempo de processamento.

Capítulo 6

Conclusões

Seguindo a prática adotada nas seções de Metodologia e Resultados, dividimos as conclusões do trabalho em duas partes, sendo a primeira relativa à Fase 1 - Geração da base de dados e a segunda relativa à Fase 2 - Desenvolvimento do modelo CNN. Na sequência, apresentamos as sugestões de trabalhos futuros.

Durante o projeto, foram calculados o impacto de diversos tipos de degradações, em modelos de reconhecimento facial considerados estado da arte. O objetivo deste trabalho é identificar com precisão, quanto de desempenho determinado modelo perde ao lidar com imagens degradadas como entrada para o algoritmo.

Em seguida, treinamos diversos modelos de classificação de imagens, a fim de prever qual a degradação presente em uma imagem da face. Com isto, torna-se possível identificar automaticamente a degradação presente e, com base no estudo anterior, prever qual o impacto que o par de imagens utilizado pode causar em determinado algoritmo de reconhecimento facial.

6.1 Fase 1 - Avaliação da Influência das Degradações em Modelos Profundos

A contribuição principal deste trabalho, foi a proposta de uma abordagem para quantificar o impacto de 14 tipos de degradações em 24 diferentes pipelines (8 algoritmos de reconhecimento facial e 3 algoritmos de detecção facial) de sistemas de reconhecimento facial, onde todos foram submetidos a 4 conjuntos de dados de imagens faciais. Além disso, foram gerados 3 tipos de pares de imagens para suportar diferentes cenários de aplicação, sempre buscando simular situações reais.

6.1.1 Reconhecimento Facial

Ao analisar os resultados obtidos para os algoritmos de reconhecimento facial, verificou-se que a métrica acurácia não correspondia à melhor forma de analisar o comportamento dos algoritmos, tendo em vista que a quantidade de pares negativos é muito superior à quantidade de pares positivos, e a referida métrica leva em consideração todos os resultados gerados. Desta forma, optou-se por utilizar a métrica *recall*, a qual, para melhor entendimento do experimento, foi analisada em relação aos pares positivos e negativos (exceto pelo Par 1 porque não possui pares negativos).

Todos os resultados demonstraram, como esperado, uma diminuição gradual na curva de *recall* para pares positivos para a geração dos Pares 1 e 2 - Imagem Padrão (não degradada) x Cópia de imagem padrão (degradada) e Imagem Padrão (não degradada) x Imagem questionada (degradada). Para o Par 3 – imagem padrão (degradada) x imagem questionada (degradada), foi identificada uma tendência extremamente perigosa, possivelmente causando erros de interpretação dos resultados. Esse comportamento foi observado para a maioria dos *pipelines* de reconhecimento facial testados.

Conforme descrito no trabalho apresentado, quando duas imagens degradadas são submetidas ao sistema de reconhecimento facial além de um determinado ponto de intensidade de degradação, o algoritmo tende a inferir que as duas imagens pertencem à mesma pessoa, mesmo sendo indivíduos diferentes. Em outras palavras, independentemente de o par de imagens ser positivo ou negativo, o algoritmo produz um resultado positivo.

Num cenário de investigação criminal, este comportamento pode levar a que um suspeito e um cidadão inocente sejam considerados a mesma pessoa, induzindo ao erro o perito e, consequentemente, as demais entidades do sistema de justiça criminal. Este mal-entendido específico pode resultar em violência material e moral irreparável contra um indivíduo.

Nesta observação, é importante estudar mecanismos no futuro, como algoritmos de avaliação de qualidade facial, para identificar o momento exato de cada algoritmo e tipo de degradação quando a curva muda de uma tendência descendente para uma tendência ascendente, para evitar erros de interpretação por especialistas.

As degradações que apresentaram mais impacto no desempenho dos pipelines foram, conforme esperado, as degradações em sequência, a saber: Borramento Gaussiano → Escurecimento → Redução de Tamanho → Compressão JPEG, Borramento Gaussiano → Brilho → Compressão JPEG e Borramento Gaussiano → Escurecimento → Redução de Tamanho.

Por outro lado, as degradações que tiveram menor impacto nos pipelines de reconhecimento facial, também conforme esperado, foram as degradações simples, a saber: Compressão JPEG, Escurecimento e Brilho.

Do ponto de vista do modelo, o modelo de reconhecimento facial mais resiliente que exibiu consistentemente o melhor desempenho inicial e degradação mínima de desempenho sob essas degradações foi o VGGFace. O modelo performou com acurácia média de 94% para o nível 3 e, 34% para o nível 6, considerando todas as degradações estudadas.

6.1.2 Detecção Facial

Em relação à análise dos algoritmos de detecção facial, mtcnn e retinaface se provaram mais robustos frente ao dlib na maioria dos experimentos. Das 14 (quatroze) degradações estudadas, o dlib performou em último lugar em 9 (nove) delas. Ao analisar individualmente cada algoritmo, mtcnn apresentou mais robustez nas degradações simples de Compressão JPEG e ruído Sal e Pimenta. Já algoritmo retinaface apresentou robustez nas degradações de Brilho e Redução de Tamanho.

Nas degradações em sequência, é preciso avaliar cada caso para verificar o desempenho, visto que determinada degradação pode favorecer ou prejudicar determinado algoritmo de detecção facial. Mas geralmente, as degradações em que estavam presentes as degradações de Compressão JPEG e ruído Sal e Pimenta, o algoritmo mtcnn se sobressaiu em relação ao retinaface. Em compensação, o retinaface se sobressaiu nas degradações onde havia Redução de Tamanho e Brilho.

Desta forma, ficou evidente que a escolha do algoritmo de detecção facial, conforme o tipo de degradação presente na imagem, possui alta relevância no resultado final do sistema de reconhecimento facial, podendo ser o fator decisivo para um resultado correto.

6.2 Fase 2 - Definição de Modelos para Detecção de Degradações

O treinamento realizado utilizando aprendizagem por transferência de conhecimento (do inglês *transfer learning*) geralmente se mostrou mais rápido do que o treinamento do zero (do inglês *from scratch*), empregando o mesmo modelo e hiperparâmetros e base de dados (*dataset*). Isto sugere que, além de convergir mais rapidamente, a abordagem por *transfer learning* também estressa menos o *hardware*.

Vale ressaltar que a precisão obtida com a abordagem por aprendizagem por transferência (*transfer learning*) nem sempre é superior à abordagem do zero (*from scratch*). Um exemplo disso é a precisão alcançada em determinados modelos, como Inception-V3, Inception-V4, Xception-71 e MobileNet-V2.

Outro ponto que merece atenção é que o modelo MobileNet-v2-100, apesar de ter sido projetado com uma arquitetura mais leve e adaptada para dispositivos móveis com menor

poder de processamento, obteve alta precisão, competindo com a maioria dos outros modelos do experimento.

Analisando a curva de erro dos 3 principais modelos, pode-se observar que os modelos DenseNet-201 (TL) e ResNet-152 (TL) obtiveram um treinamento mais suave, enquanto o modelo Inception-V3 (FS) exibiu um processo de treinamento com mais picos. Este fenômeno é possivelmente atribuído ao fato de os dois primeiros modelos utilizarem técnicas de aprendizagem por transferência, enquanto o modelo Inception foi treinado do zero. Neste ponto, mais estudos poderiam ser realizados para melhor compreender esse comportamento.

Continuando a analisar os três principais modelos, observou-se que o Inception-v3 (FS) completou 100 épocas de treinamento em aproximadamente 2,3 dias, enquanto o DenseNet-201 (TL) levou 4,3 dias e o ResNet-152 (TL) levou 6,7 dias. Essas redes passaram por treinamento utilizando um conjunto de dados de 787.363 imagens. Desta forma, é evidente que o modelo Inception-v3 (FS) treinou significativamente mais rápido, exigindo quase metade do tempo do DenseNet-201 (TL) e quase um terço do tempo de uso da GPU em comparação com o ResNet-152 (TL).

Exceto pelo modelo ResNet-18 (FS), todos os outros modelos atingiram níveis satisfatórios de precisão ao final das 100 épocas de treinamento. Isto indica que independentemente do modelo estado da arte utilizado, a tarefa de classificar a degradação em uma imagem de face pode ser realizada por várias arquiteturas com uma precisão muito próxima.

6.3 Trabalhos Futuros

Nesta parte, descrevemos alguns pontos que podem ser aprofundados em trabalhos futuros.

Devido ao alto grau de inovação na área, novos modelos surgem com frequência, sendo inviável abordar todos os modelos desenvolvidos recentemente. Assim, a fim de expandir o estudo e implementar novas comparações, novos modelos podem ter seus gráficos de impacto gerados e, consequentemente, comparados com os modelos abordados neste trabalho. Ainda neste sentido, novos tipos de degradações podem ser adicionados, possibilitando identificar a robustez dos algoritmos frente a novos problemas encontrados no cenário real.

Atualmente, uma revisão da "ISO 29794-5 - Part 5: Face image data" encontra-se em desenvolvimento. Esta ISO busca estabelecer aspectos da qualidade da imagem da face, especificar termos e definições, e padronizar os escores relativos à qualidade da face. Tendo em vista a proximidade do assunto abordado nesta ISO com o atual trabalho, é

possível sugerir que alguma mudança nos tipos de degradações utilizadas neste trabalho possa ocorrer para melhor adequar este estudo à referida ISO.

Durante o treinamento dos modelos de classificação de imagens, pudemos verificar alguns pontos que também podem ser melhor explorados no futuro. No tocante às abordagens utilizadas com transferência de conhecimento e do zero, estudos podem ser realizados no sentido de melhor entender a diferença de tempo de treinamento para a mesma quantidade de épocas, a suavidade das curvas de erro, o estresse imposto ao *hardware* para treinamento, e a acurácia final obtida.

O fato da ResNet-152 (FS) (88,75%) ter atingido uma acurácia inferior à ResNet-50 (FS) (90,07%), pode ser melhor averiguado. Somando-se a isso, a mesma arquitetura (ResNet-152), quando treinada com uso de *transfer learning*, obteve acurácia entre os 3 melhores modelos do experimento (94,15%). Neste sentido, especialmente na arquitetura ResNet-152, a escolha na abordagem de treinamento utilizada resultou em uma grande diferença entre as acurácias obtidas (quase 6%).

Aproveitando este estudo, ao estimar com certa precisão os tipos de degradações presentes em uma imagem (simples e em sequência), pode-se treinar um modelo baseado em *diffusors* a fim de que este remova determinado tipo (ou conjunto) de degradação. Este modelo pode fazer uso de prompt de texto para direcionar a remoção do conjunto de degradação. Como resultado, o modelo seria capaz de entregar como output uma imagem livre (ou próximo a isto) de degradação.

6.4 Publicações Realizadas

Neste capítulo, descreveremos as publicações realizadas com os resultados dos experimentos e as análises obtidas.

Como consequência da pesquisa realizada, dois artigos científicos foram elaborados, sendo um referente à Fase 1 do projeto e outro referente à Fase 2. Desta forma, os resultados foram divididos, mantendo a coerência das análises.

6.4.1 Fase 1 - Avaliação da Influência das Degradações em Modelos Profundos

A partir dos resultados e das conclusões referentes à Fase 1, foi escrito o artigo *Quantifying the impact of image degradation on Deep Learning models in face recognition systems* (Figura 6.1), o qual foi submetido, publicado e apresentado presencialmente no ENIAC - 20º Encontro Nacional de Inteligência Artificial e Computacional (Figura 6.3), realizado na cidade de Belo Horizonte, no mês de setembro de 2023.

6.4.2 Fase 2 - Definição de Modelos para Detecção de Degradações

Com base nos resultados e nas conclusões realizadas na Fase 2 do projeto, escrevemos o artigo *Face Image Quality Assessment: Unveiling Degradations with Deep Learning Models* (Figura 6.2), o qual foi submetido em setembro de 2023 para o VISAPP - 19º Conferência Internacional em Computação Visual, Teoria e Aplicações de Imagens e Computação Gráfica (Figura 6.4), a ser realizada na cidade de Roma, Itália, no mês de fevereiro de 2024. Devido ao cronograma do evento, até o presente momento não obtivemos retorno quanto à aceitação ou recusa do artigo. A previsão de resposta é dia 5 de dezembro de 2023.

Quantifying the impact of image degradation on Deep Learning models in face recognition systems

Leandro Dias Carneiro[1], **Flavio de Barros Vidal**[2]

[1]Criminalistics Institute of the Federal District Civil Police – Brasília – DF – Brazil
[2]Dep. of Computer Science – University of Brasília – Brasília – DF – Brazil

leandro.carneiro@pcdf.df.gov.br, fbvidal@unb.br

Abstract. *Significant advancements in computer vision, particularly in facial recognition systems, have been witnessed in recent years. However, it is imperative to comprehend how these systems perform under real-world conditions, specifically when confronted with degraded images. This paper presents a comprehensive analysis of the impact of image degradation on facial recognition systems that rely on deep neural networks. The study evaluates three facial detection algorithms and eight facial recognition algorithms, with experiments conducted on four diverse datasets. A total of 14 types of image degradations, encompassing pure and mixed variations, were employed at six different intensity levels. Three distinct types of image pairs were generated to encompass various scenarios. The primary objective of this research is to enhance the understanding and assessment of facial recognition system outcomes, thereby strengthening the overall analysis of these systems. On average, the models had a minimum impact of 17% and a maximum of 43% for the datasets used in the experiment.*

Figura 6.1: *Abstract* do artigo apresentado no ENIAC.

Face Image Quality Assessment: Unveiling Degradations with Deep Learning Models

Keywords: Image Degradation, Face Recognition, Face Image Degradation, Face Image Quality.

Abstract: Recent years have seen remarkable progress in computer vision, mainly in facial recognition systems using deep models. However, understanding the real-world performance of these systems, especially with suboptimal and degraded images, is crucial for many scenarios, such as forensics applications and security systems. This study introduces a comprehensive exploration and evaluation involving ten state-of-the-art (SOTA) models of convolutional neural networks used in face recognition tasks, trained from scratch and using transfer learning, resulting in 20 different trained models. The study identifies degradation type and intensity in facial images and determines the top models for this identification. Fourteen image degradation types, spanning pure to mixed variations, were tested at least across six intensity levels. Following training, the most efficient models achieved an impressive 94% accuracy, while the least performing model reached 71% accuracy in an overall evaluation.

Figura 6.2: *Abstract* do artigo submetido no VISAPP.

Figura 6.3: ENIAC

Figura 6.4: VISAPP.

Referências

[1] Adjabi, Insaf, Abdeldjalil Ouahabi, Amir Benzaoui e Abdelmalik Taleb-Ahmed: *Past, present, and future of face recognition: A review.* Electronics, 9(8):1188, 2020. xi, 1, 2, 8, 16

[2] Schlett, Torsten, Christian Rathgeb, Olaf Henniger, Javier Galbally, Julian Fierrez e Christoph Busch: *Face image quality assessment: A literature survey.* ACM Computing Surveys (CSUR), 54(10s):1–49, 2022. xi, xii, 3, 9, 10, 37, 38, 39, 40, 49

[3] Fuad, Md Tahmid Hasan, Awal Ahmed Fime, Delowar Sikder, Md Akil Raihan Iftee, Jakaria Rabbi, Mabrook S Al-Rakhami, Abdu Gumaei, Ovishake Sen, Mohtasim Fuad e Md Nazrul Islam: *Recent advances in deep learning techniques for face recognition.* IEEE Access, 9:99112–99142, 2021. xi, 9, 10, 11, 12, 13, 15, 34, 35, 36, 48

[4] Wang, Mei e Weihong Deng: *Deep face recognition: A survey.* Neurocomputing, 429:215–244, 2021. xi, 1, 3, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 29, 33, 34, 38, 39, 48, 49

[5] Tsang, Sik Ho: *Twards data science - resnet.* `https://towardsdatascience.com/review-resnet-winner-of-ilsvrc-2015-image-classification-localization-detection-` xi, 16

[6] Aghdam, Omid Abdollahi e Hazım Kemal Ekenel: *Robust deep learning features for face recognition under mismatched conditions.* Em *2018 26th Signal Processing and Communications Applications Conference (SIU)*, páginas 1–4. IEEE, 2018. xi, 15, 16, 17

[7] He, Kaiming, Xiangyu Zhang, Shaoqing Ren e Jian Sun: *Deep residual learning for image recognition.* Em *Proceedings of the IEEE conference on computer vision and pattern recognition*, páginas 770–778, 2016. xi, 15, 17, 25, 30, 50, 55

[8] Tang, Pengjie, Hanli Wang e Sam Kwong: *G-ms2f: Googlenet based multi-stage feature fusion of deep cnn for scene recognition.* Neurocomputing, 225:188–197, 2017. xi, 15, 17, 53, 54, 55

[9] Krizhevsky, Alex, Ilya Sutskever e Geoffrey E Hinton: *Imagenet classification with deep convolutional neural networks.* Em Pereira, F., C. J. C. Burges, L. Bottou e K. Q. Weinberger (editores): *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. `https://proceedings.neurips.cc/`

paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf. xi, 7, 13, 17, 53, 54, 55

[10] Simonyan, Karen e Andrew Zisserman: *Very deep convolutional networks for large-scale image recognition.* arXiv preprint arXiv:1409.1556, 2014. xi, 15, 17, 51, 54, 55

[11] Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever e Ruslan Salakhutdinov: *Dropout: a simple way to prevent neural networks from overfitting.* The journal of machine learning research, 15(1):1929–1958, 2014. xi, 13, 18, 19

[12] He, Kaiming, Xiangyu Zhang, Shaoqing Ren e Jian Sun: *Delving deep into rectifiers: Surpassing human-level performance on imagenet classification.* Em *Proceedings of the IEEE international conference on computer vision*, páginas 1026–1034, 2015. xi, 19

[13] Shorten, Connor e Taghi M Khoshgoftaar: *A survey on image data augmentation for deep learning.* Journal of big data, 6(1):1–48, 2019. xi, 13, 18, 19

[14] King, Davis E: *Dlib-ml: A machine learning toolkit.* The Journal of Machine Learning Research, 10:1755–1758, 2009. xi, 3, 20, 25, 60

[15] Serengil, Sefik Ilkin: *Serengil - deepface.* https://github.com/serengil/deepface. xi, 20, 21, 23, 24, 25, 26, 27, 28

[16] Deng, Jiankang, Jia Guo, Evangelos Ververas, Irene Kotsia e Stefanos Zafeiriou: *Retinaface: Single-shot multi-level face localisation in the wild.* Em *CVPR*, 2020. xi, 3, 20, 21, 22, 60

[17] Deng, Jiankang, Jia Guo, Niannan Xue e Stefanos Zafeiriou: *Arcface: Additive angular margin loss for deep face recognition.* Em *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, páginas 4690–4699, 2019. xi, xii, 3, 11, 28, 29, 30, 36, 37, 60

[18] Boutros, Fadi, Marco Huber, Patrick Siebke, Tim Rieber e Naser Damer: *Sface: Privacy-friendly and accurate face recognition using synthetic data.* arXiv preprint arXiv:2206.10520, 2022. xii, 3, 30, 31, 60

[19] *Tensorflow addons losses: Tripletsemihardloss.* https://www.tensorflow.org/addons/tutorials/losses_triplet. xii, 35

[20] Manwatkar, Charudatta: *How to choose a loss function for face recognition.* https://neptune.ai/blog/how-to-choose-loss-function-for-face-recognition. xii, 35, 36, 37

[21] Meng, Qiang, Shichao Zhao, Zhida Huang e Feng Zhou: *Magface: A universal representation for face recognition and quality assessment.* CoRR, abs/2103.06627, 2021. https://arxiv.org/abs/2103.06627. xii, 38, 39, 40

[22] Rodríguez, Joaquín Salas, Flavio De Barros Vidal e Francisco Martinez-Trinidad: *Deep learning: Current state*. IEEE Latin America Transactions, 17(12):1925–1945, 2019. 1

[23] LeCun, Yann, Yoshua Bengio *et al.*: *Convolutional networks for images, speech, and time series*. The handbook of brain theory and neural networks, 3361(10):1995, 1995. 1, 7, 12

[24] Coşkun, Musab, Ayşegül Uçar, Özal Yildirim e Yakup Demir: *Face recognition based on convolutional neural network*. Em *2017 International Conference on Modern Electrical and Energy Systems (MEES)*, páginas 376–379. IEEE, 2017. 1

[25] Abade, André, Paulo Afonso Ferreira e Flavio de Barros Vidal: *Plant diseases recognition on images using convolutional neural networks: A systematic review*. Computers and Electronics in Agriculture, 185:106125, 2021, ISSN 0168-1699. `https://www.sciencedirect.com/science/article/pii/S0168169921001435`. 1

[26] Freitas Pereira, Tiago de, Dominic Schmidli, Yu Linghu, Xinyi Zhang, Sébastien Marcel e Manuel Günther: *Eight years of face recognition research: Reproducibility, achievements and open issues*. arXiv, (2208.04040), 2022. 1, 2, 3, 8, 29, 36

[27] Bansal, Ankan, Rajeev Ranjan, Carlos D Castillo e Rama Chellappa: *Deep cnn face recognition: Looking at the past and the future*. Deep Learning-Based Face Analytics, páginas 1–20, 2021. 1, 2, 8, 10, 11

[28] Smith, Marcus, Monique Mann e Gregor Urbas: *Biometrics, crime and security*. Routledge, 2018. 2, 3, 4

[29] GRECO, Rogério e ANTONIO MARTINS: *Direito penal*. Vol I. Ímpetus, 2016. 2

[30] Lopes Jr, Aury: *Direito processual penal*. Saraiva Educação SA, 2018. 2

[31] Grm, Klemen, Vitomir Štruc, Anais Artiges, Matthieu Caron e Hazım K Ekenel: *Strengths and weaknesses of deep learning models for face recognition against image degradations*. Iet Biometrics, 7(1):81–89, 2018. 2, 3, 40, 43, 54, 57, 61

[32] *Iso 19794-5 documentation. available: https://www.iso.org/standard/50867.html*. `https://www.iso.org/standard/50867.html`. 2, 61

[33] *Icao documentation. available: https://www.icao.int/publications/pages/publication.aspx?docnum*
`https://www.icao.int/publications/pages/publication.aspx?docnum=9303`.
2

[34] Zeinstra, Chris G, Didier Meuwly, A Cc Ruifrok, R Nj Veldhuis e Lieuwe Jan Spreeuwers: *Forensic face recognition as a means to determine strength of evidence: a survey*. Forensic Sci Rev, 30(1):21–32, 2018. 2, 64

[35] Huang, Gary B., Manu Ramesh, Tamara Berg e Erik Learned-Miller: *Labeled faces in the wild: A database for studying face recognition in unconstrained environments*. Relatório Técnico 07-49, University of Massachusetts, Amherst, October 2007. 2, 23, 25, 30, 47, 53, 54, 56, 63

[36] Smith, Marcus e Seumas Miller: *The ethical application of biometric facial recognition technology.* Ai & Society, páginas 1–9, 2022. 3, 4

[37] Parkhi, Omkar M., Andrea Vedaldi e Andrew Zisserman: *Deep face recognition.* Em *Proceedings of the British Machine Vision Conference (BMVC)*, páginas 41.1–41.12. BMVA Press, September 2015, ISBN 1-901725-53-7. `https://dx.doi.org/10.5244/C.29.41`. 3, 26, 27, 53, 54

[38] Schroff, Florian, Dmitry Kalenichenko e James Philbin: *Facenet: A unified embedding for face recognition and clustering.* 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jun 2015. `http://dx.doi.org/10.1109/CVPR.2015.7298682`. 3, 26, 34, 60

[39] Amos, Brandon, Bartosz Ludwiczuk e Mahadev Satyanarayanan: *Openface: A general-purpose face recognition library with mobile applications.* Relatório Técnico, CMU-CS-16-118, CMU School of Computer Science, 2016. 3, 27, 28, 60

[40] Taigman, Yaniv, Ming Yang, Marc'Aurelio Ranzato e Lior Wolf: *Deepface: Closing the gap to human-level performance in face verification.* Em *Proceedings of the IEEE conference on computer vision and pattern recognition*, páginas 1701–1708, 2014. 3, 23, 24, 60

[41] Sun, Yi, Yuheng Chen, Xiaogang Wang e Xiaoou Tang: *Deep learning face representation by joint identification-verification.* Advances in neural information processing systems, 27, 2014. 3, 24, 60

[42] Sun, Yi, Yuheng Chen, Xiaogang Wang e Xiaoou Tang: *Deep learning face representation by joint identification-verification.* Em *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, página 1988–1996, Cambridge, MA, USA, 2014. MIT Press. 3

[43] Zhang, Kaipeng, Zhanpeng Zhang, Zhifeng Li e Yu Qiao: *Joint face detection and alignment using multitask cascaded convolutional networks.* IEEE signal processing letters, 23(10):1499–1503, 2016. 3, 20, 21, 60

[44] Bledsoe, W. W.: *The model method in facial recognition.* Technical report PRI 15, Panoramic Research, Inc., 1964. 6

[45] Bledsoe, W. W. e H. Chan: *A man-machine facial recognition system-some preliminary results.* Technical report PRI 19A, Panoramic Research, Inc., 1965. 6

[46] Bledsoe, W. W.: *Man-machine facial recognition: Report on a large-scale experiment.* Technical report PRI 22, Panoramic Research, Inc., 1966. 6

[47] Bledsoe, W. W.: *Semiautomatic facial recognition.* Technical report SRI Project 6693, Stanford Research Institute, 1968. 6

[48] Kanade, Takeo: *Picture processing system by computer complex and recognition of human faces*, November 1973. 6

[49] Turk, Matthew e Alex Pentland: *Eigenfaces for recognition*. J. Cognitive Neuroscience, 3(1):71–86, janeiro 1991, ISSN 0898-929X. https://doi.org/10.1162/jocn.1991.3.1.71. 7, 48

[50] Etemad, Kamran e Rama Chellappa: *Discriminant analysis for recognition of human face images*. J. Opt. Soc. Am. A, 14(8):1724–1733, Aug 1997. http://josaa.osa.org/abstract.cfm?URI=josaa-14-8-1724. 7

[51] Russakovsky, Olga, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein *et al.*: *Imagenet large scale visual recognition challenge*. International Journal of Computer Vision, 115(3):211–252, 2015. 7, 15, 54, 55, 68

[52] Du, Hang, Hailin Shi, Dan Zeng, Xiao Ping Zhang e Tao Mei: *The elements of end-to-end deep face recognition: A survey of recent advances*. ACM Computing Surveys (CSUR), 54(10s):1–42, 2022. 8, 9, 10, 11, 12, 29, 43

[53] *Iso 39794-5 documentation. available: https://www.iso.org/standard/72156.html*. https://www.iso.org/standard/72156.html. 9, 38

[54] Rasamoelina, Andrinandrasana David, Fouzia Adjailia e Peter Sinčák: *A review of activation function for artificial neural network*. Em *2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, páginas 281–286. IEEE, 2020. 11, 19

[55] Wang, Hao, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li e Wei Liu: *Cosface: Large margin cosine loss for deep face recognition*. Em *Proceedings of the IEEE conference on computer vision and pattern recognition*, páginas 5265–5274, 2018. 11, 30, 36

[56] Agarap, Abien Fred: *Deep learning using rectified linear units (relu)*. arXiv preprint arXiv:1803.08375, 2018. 13, 19

[57] Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville e Yoshua Bengio: *Generative adversarial networks*. Communications of the ACM, 63(11):139–144, 2020. 18

[58] Ioffe, Sergey e Christian Szegedy: *Batch normalization: Accelerating deep network training by reducing internal covariate shift*. Em *International conference on machine learning*, páginas 448–456. PMLR, 2015. 18

[59] Glorot, Xavier, Antoine Bordes e Yoshua Bengio: *Deep sparse rectifier neural networks*. Em *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, páginas 315–323. JMLR Workshop and Conference Proceedings, 2011. 19

[60] Guo, Jia e Jiankang Deng: *deepinsight/insightface: Face analysis project on mxnet*, 2020. 21

[61] Taigman, Yaniv, Ming Yang, Marc'Aurelio Ranzato e Lior Wolf: *Deepface: Closing the gap to human-level performance in face verification*. Em *Proceedings of the IEEE conference on computer vision and pattern recognition*, páginas 1701–1708, 2014. 23

[62] King, Davis E.: *Dlib documentation. available: http://dlib.net/*. `http://dlib.net/`. 25

[63] Ng, Hong Wei e Stefan Winkler: *A data-driven approach to cleaning large face datasets*. Em *2014 IEEE International Conference on Image Processing (ICIP)*, páginas 343–347. IEEE, 2014. 25

[64] Cao, Qiong, Li Shen, Weidi Xie, Omkar M Parkhi e Andrew Zisserman: *Vggface2: A dataset for recognizing faces across pose and age*. Em *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, páginas 67–74. IEEE, 2018. 25, 30

[65] Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke e Andrew Rabinovich: *Going deeper with convolutions*. Em *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, páginas 1–9, 2015. 26, 51, 55

[66] Guo, Yandong, Lei Zhang, Yuxiao Hu, Xiaodong He e Jianfeng Gao: *Ms-celeb-1m: A dataset and benchmark for large-scale face recognition*. Em *European Conference on Computer Vision*, páginas 87–102. Springer, 2016. 30

[67] Kemelmacher-Shlizerman, Ira, Steven M Seitz, Daniel Miller e Evan Brossard: *The megaface benchmark: 1 million faces for recognition at scale*. Em *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, páginas 4873–4882, 2016. 30

[68] Gou, Mengran, Srikrishna Karanam, Wenqian Liu, Octavia Camps e Richard J Radke: *Dukemtmc4reid: A large-scale multi-camera person re-identification dataset*. Em *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, páginas 10–19, 2017. 30

[69] Goodfellow, Ian, Yoshua Bengio e Aaron Courville: *Deep learning*. MIT press, 2016. 30

[70] Karras, Tero, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen e Timo Aila: *Training generative adversarial networks with limited data*. Advances in Neural Information Processing Systems, 33:12104–12114, 2020. 30

[71] Srivastava, Yash, Vaishnav Murali e Shiv Ram Dubey: *A performance comparison of loss functions for deep face recognition*. CoRR, abs/1901.05903, 2019. `http://arxiv.org/abs/1901.05903`. 33, 34, 36

[72] Liu, Weiyang, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj e Le Song: *Sphereface: Deep hypersphere embedding for face recognition*. Em *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, páginas 6738–6746, 2017. 33, 35

[73] Wang, Feng, Jian Cheng, Weiyang Liu e Haijun Liu: *Additive margin softmax for face verification*. IEEE Signal Processing Letters, 25(7):926–930, 2018. 33

[74] Damera-Venkata, Niranjan, Thomas D Kite, Wilson S Geisler, Brian L Evans e Alan C Bovik: *Image quality assessment based on a degradation model*. IEEE transactions on image processing, 9(4):636–650, 2000. 37, 39

[75] *Iso 29794-5 documentation. available: https://www.iso.org/standard/50912.html*. `https://www.iso.org/standard/50912.html`. 38, 61

[76] Alonso-Fernandez, Fernando, Julian Fiérrez e Javier Ortega-Garcia: *Quality measures in biometric systems*. CoRR, abs/2111.08704, 2021. `https://arxiv.org/abs/2111.08704`. 39, 40

[77] Ashraf, Md.Naseem: *Image degradation and noise*. `https://www.slideshare.net/NaseemAshraf/image-degradation-and-noise-by-mdnaseem-ashraf`. 40, 44, 45

[78] Karahan, Samil, Merve Kilinc Yildirum, Kadir Kirtac, Ferhat Sukru Rende, Gultekin Butun e Hazim Kemal Ekenel: *How image degradations affect deep cnn-based face recognition?* Em *2016 international conference of the biometrics special interest group (BIOSIG)*, páginas 1–5. IEEE, 2016. 40, 43, 53, 57, 61

[79] Liu, Ding, Bowen Cheng, Zhangyang Wang, Haichao Zhang e Thomas S Huang: *Enhance visual recognition under adverse conditions via deep networks*. IEEE Transactions on Image Processing, 28(9):4401–4412, 2019. 40, 43, 44, 45, 46, 54, 57, 61

[80] Roy, Prasun, Subhankar Ghosh, Saumik Bhattacharya e Umapada Pal: *Effects of degradations on deep neural network architectures*. arXiv preprint arXiv:1807.10108, 2018. 40, 43, 44, 45, 54, 55, 57, 61

[81] Pei, Yanting, Yaping Huang, Qi Zou, Hao Zang, Xingyuan Zhang e Song Wang: *Effects of image degradations to cnn-based image classification*. arXiv preprint arXiv:1810.05552, 2018. 40, 43, 45, 55, 57, 61

[82] Aljarrah, Inad A: *Effect of image degradation on performance of convolutional neural networks*. International Journal of Communication Networks and Information Security, 13(2):215–219, 2021. 40, 45, 55, 57, 61

[83] Thomaz, Dr. Carlos Eduardo: *Fei face database*. Relatório Técnico, University of São Bernardo do Campo, 2006. 47, 63

[84] Grgic, Mislav, Kresimir Delac e Sonja Grgic: *Scface — surveillance cameras face database*. Multimedia Tools Appl., 51(3):863–879, feb 2011, ISSN 1380-7501. `https://doi.org/10.1007/s11042-009-0417-2`. 48, 63

[85] Burton, A Mike, David White e Allan McNeill: *The glasgow face matching test*. Behavior research methods, 42(1):286–291, 2010. 48, 63

[86] Huang, Gao, Zhuang Liu, Laurens Van Der Maaten e Kilian Q Weinberger: *Densely connected convolutional networks*. Em *Proceedings of the IEEE conference on computer vision and pattern recognition*, páginas 4700–4708, 2017. 50

[87] Chollet, François: *Xception: Deep learning with depthwise separable convolutions*. Em *Proceedings of the IEEE conference on computer vision and pattern recognition*, páginas 1251–1258, 2017. 51

[88] Howard, Andrew G., Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, M. Andreetto e Hartwig Adam: *Mobilenets: Efficient convolutional neural networks for mobile vision applications*. ArXiv, abs/1704.04861, 2017. 52, 54, 55

[89] Sandler, M., Andrew G. Howard, Menglong Zhu, A. Zhmoginov e Liang Chieh Chen: *Mobilenetv2: Inverted residuals and linear bottlenecks*. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, páginas 4510–4520, 2018. 52, 54, 55

[90] Iandola, Forrest N, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally e Kurt Keutzer: *Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size*. arXiv preprint arXiv:1602.07360, 2016. 54

[91] Krizhevsky, Alex, Geoffrey Hinton *et al.*: *Learning multiple layers of features from tiny images*. 2009. 54

[92] Zhang, Xiao, Lei Zhang, Xin Jing Wang e Heung Yeung Shum: *Finding celebrities in billions of web images*. IEEE Transactions on Multimedia, 14(4):995–1007, 2012. 54

[93] Yang, Shuo, Ping Luo, Chen Change Loy e Xiaoou Tang: *Wider face: A face detection benchmark*. Em *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, páginas 5525–5533, 2016. 54

[94] Netzer, Yuval, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu e Andrew Y Ng: *Reading digits in natural images with unsupervised feature learning*. 2011. 54

[95] Griffin, G: *a. holub, and p. perona. caltech-256 object category dataset*. Relatório Técnico, Caltech mimeo, 11 (1): 20, 2007. 55

[96] Everingham, Mark, Luc Van Gool, Christopher KI Williams, John Winn e Andrew Zisserman: *The pascal visual object classes (voc) challenge*. International journal of computer vision, 88:303–308, 2009. 55

[97] Singh, Praneet, Haoyu Chen, Edward J Delp e Amy R Reibman: *Evaluating image quality estimators for face matching*. Em *2022 IEEE 5th International Conference on Multimedia Information Processing and Retrieval (MIPR)*, páginas 204–209. IEEE, 2022. 56

[98] Hernandez-Ortega, J., Javier Galbally, Julian Fierrez e Laurent Beslay: *Biometric quality: Review and application to face recognition with faceqnet*. ArXiv, abs/2006.03298, 2020. 56

[99] Terhörst, Philipp, Jan Niklas Kolf, Naser Damer, Florian Kirchbuchner e Arjan Kuijper: *Ser-fiq: Unsupervised estimation of face image quality based on stochastic embedding robustness*. Em *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, páginas 5650–5659, 2020. 56

[100] Ou, Fu Zhao, Xingyu Chen, Ruixin Zhang, Yuge Huang, Shaoxin Li, Jilin Li, Yong Li, Liujuan Cao e Yuan Gen Wang: *Sdd-fiqa: unsupervised face image quality assessment with similarity distribution distance*. Em *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, páginas 7670–7679, 2021. 56

[101] Mittal, Anish, Anush Krishna Moorthy e Alan Conrad Bovik: *No-reference image quality assessment in the spatial domain*. IEEE Transactions on image processing, 21(12):4695–4708, 2012. 56

[102] Venkatanath, N, D Praneeth, Maruthi Chandrasekhar Bh, Sumohana S Channappayya e Swarup S Medasani: *Blind image quality evaluation using perception based features*. Em *2015 Twenty First National Conference on Communications (NCC)*, páginas 1–6. IEEE, 2015. 56

[103] Mittal, Anish, Rajiv Soundararajan e Alan C Bovik: *Making a "completely blind" image quality analyzer*. IEEE Signal processing letters, 20(3):209–212, 2012. 56

[104] Maze, Brianna, Jocelyn Adams, James A Duncan, Nathan Kalka, Tim Miller, Charles Otto, Anil K Jain, W Tyler Niggel, Janet Anderson, Jordan Cheney *et al.*: *Iarpa janus benchmark-c: Face dataset and protocol*. Em *2018 international conference on biometrics (ICB)*, páginas 158–165. IEEE, 2018. 56

[105] Atanassov, Atanas e Dimitar Pilev: *Pre-trained deep learning models for facial emotions recognition*. Em *2020 International Conference Automatics and Informatics (ICAI)*, páginas 1–6. IEEE, 2020. 60, 65

Anexo I

Resultados Completos dos Testes

Neste anexo trazemos os resultados dos experimentos realizados na Fase 1 do projeto - Geração da base de dados.

A pesquisa teve como escopo a utilização de 8 algoritmos de reconhecimento facial e 3 algoritmos de detecção facial, gerando assim 24 pipelines diferentes. Os 24 pipelines foram executados com 4 conjuntos de imagens diferentes.

I.1 Algoritmos de reconhecimento facial

Todos os experimentos são direcionados para observar e quantificar os efeitos de imagens degradadas em sistemas de reconhecimento facial. Os resultados resultantes são apresentados graficamente, estruturados da seguinte maneira:

1. Dentro de cada subseção abaixo, serão apresentados os gráficos de impacto, para cada um dos *datasets* estudados, e para cada um dos pares gerados. Desta forma, teremos 4 subseções (LFW, FEI, GUFD, SCFace) dentro de cada um dos agrupamentos acima enumerados, divididas em 3 subseções (Par 1, Par 2 e Par 3).

2. Cada imagem é dividida em três colunas, cada coluna corresponde a um modelo de detecção distinto. Dentro de cada coluna há duas linhas, referentes às amostras positivas e negativas.

Para melhor visualização, os nomes das degradações avaliadas foram abreviados como segue: Borramento Gaussiano (GBlur), Borramento de Movimento (MBlur), Brilho (Brilho), Redução de Tamanho (Down), Escurecimento (Dark), Compressão JPEG (JPEG), Ruído Gaussiano (GNoise), e Sal e Pimenta (SP).

Dataset LFW: Par 1 - Imagem padrão (não degradada) x Cópia da imagem padrão (degradada)

As imagens deste par são as imagens compreendidas da Figura I.1 até a Figura I.8

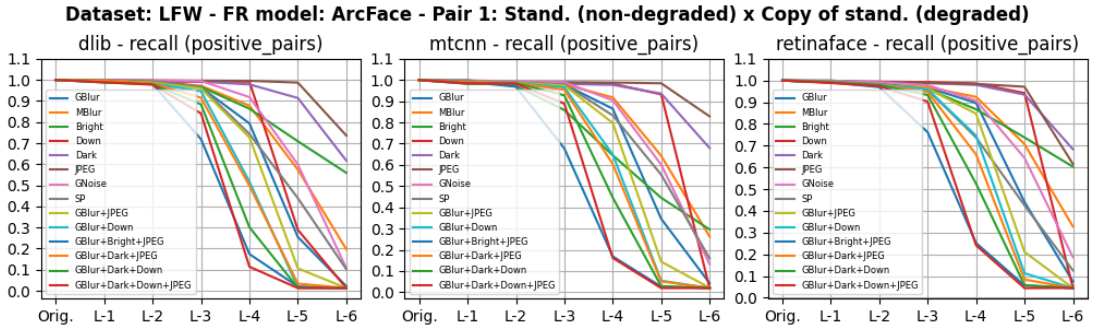


Figura I.1: Dataset LFW - Par 1 - Métrica *recall* do algoritmo ArcFace

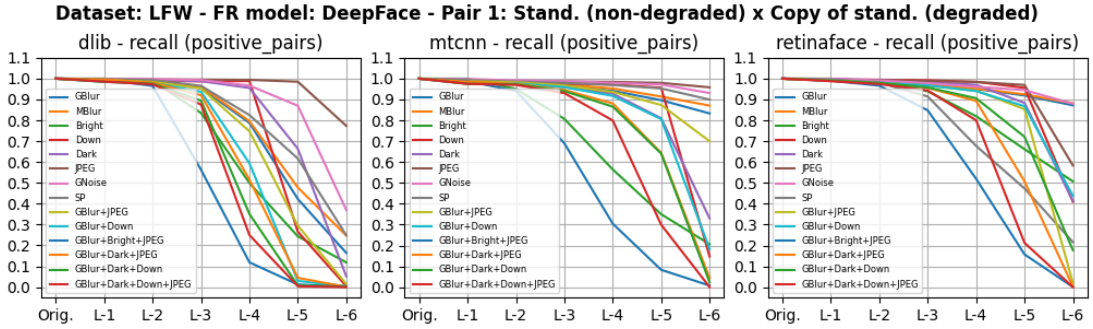


Figura I.2: Dataset LFW - Par 1 - Métrica *recall* do algoritmo DeepFace

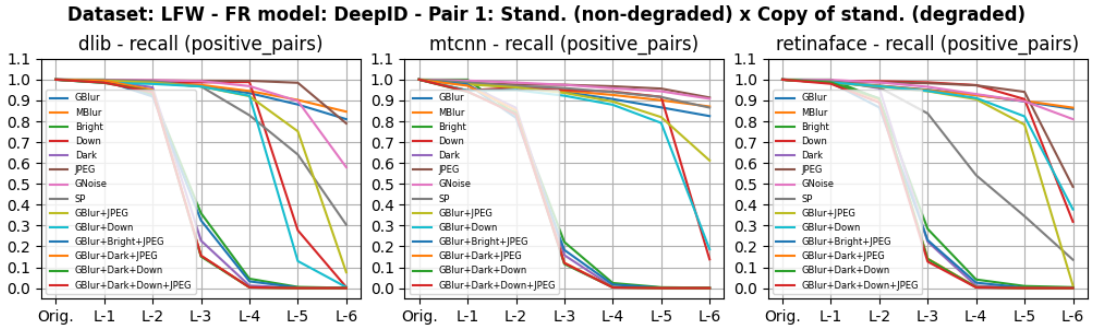


Figura I.3: Dataset LFW - Par 1 - Métrica *recall* do algoritmo DeepID

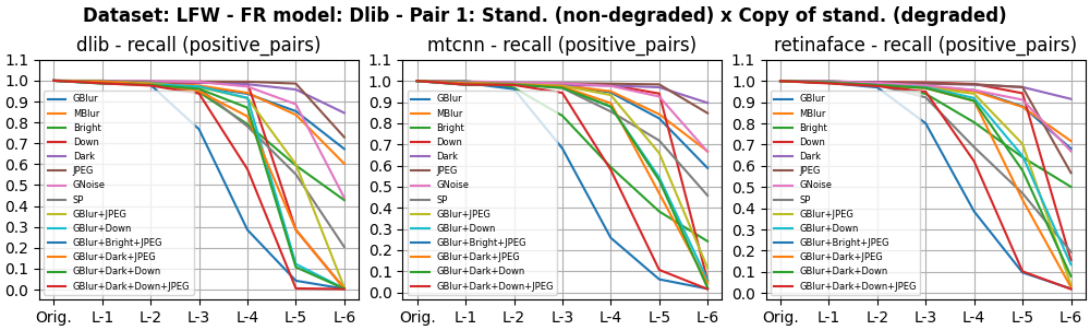


Figura I.4: Dataset LFW - Par 1 - Métrica *recall* do algoritmo Dlib

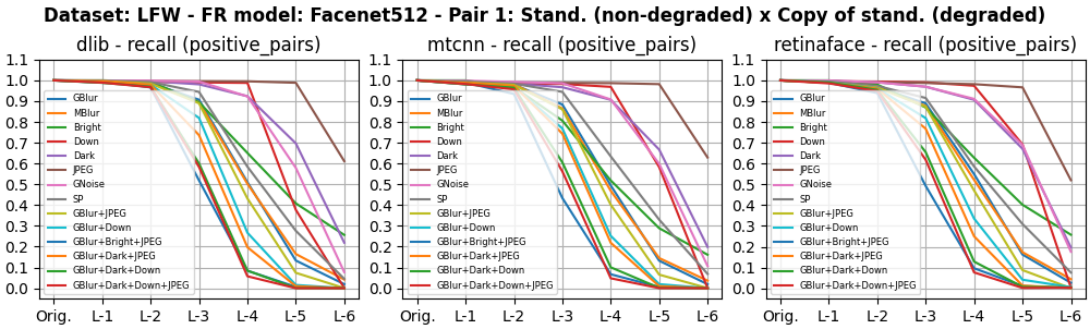


Figura I.5: Dataset LFW - Par 1 - Métrica *recall* do algoritmo FaceNet512

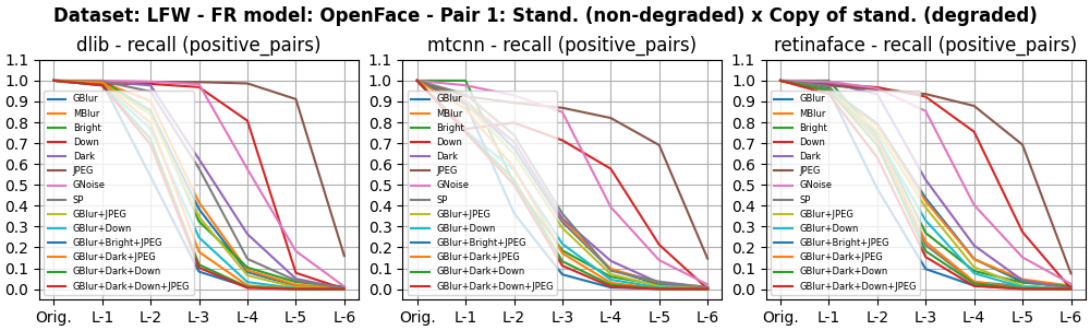


Figura I.6: Dataset LFW - Par 1 - Métrica *recall* do algoritmo OpenFace

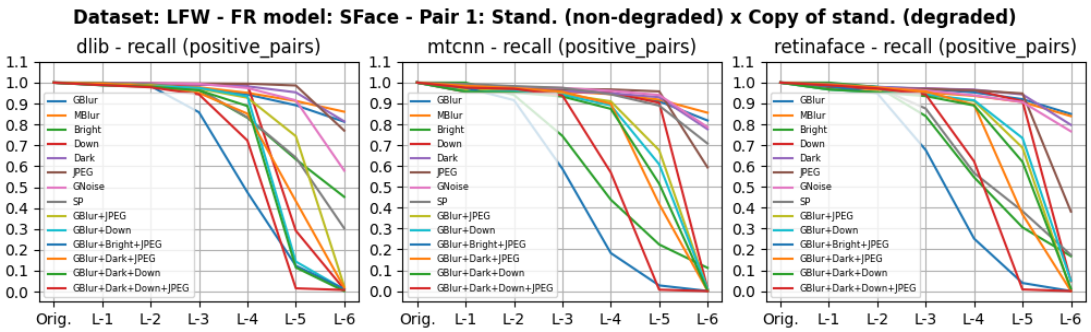


Figura I.7: Dataset LFW - Par 1 - Métrica *recall* do algoritmo SFace

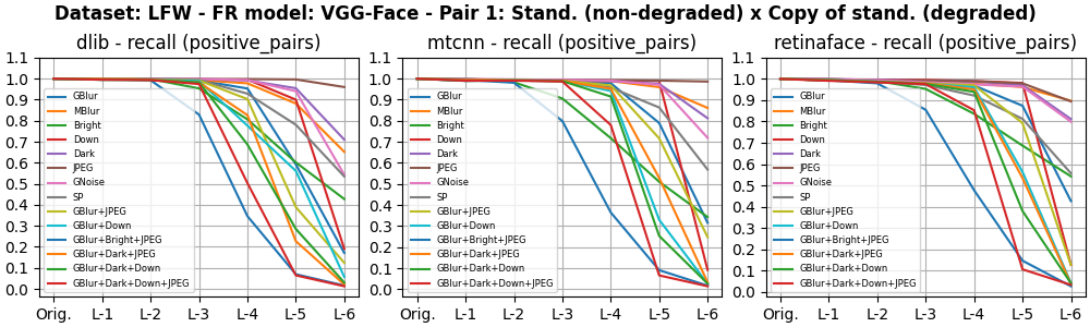


Figura I.8: Dataset LFW - Par 1 - Métrica *recall* do algoritmo VGG

Dataset LFW: Par 2 - Imagem padrão (não degradada) x Imagem questionada (degradada)

As imagens deste par são as imagens compreendidas da Figura I.9 até a Figura I.16

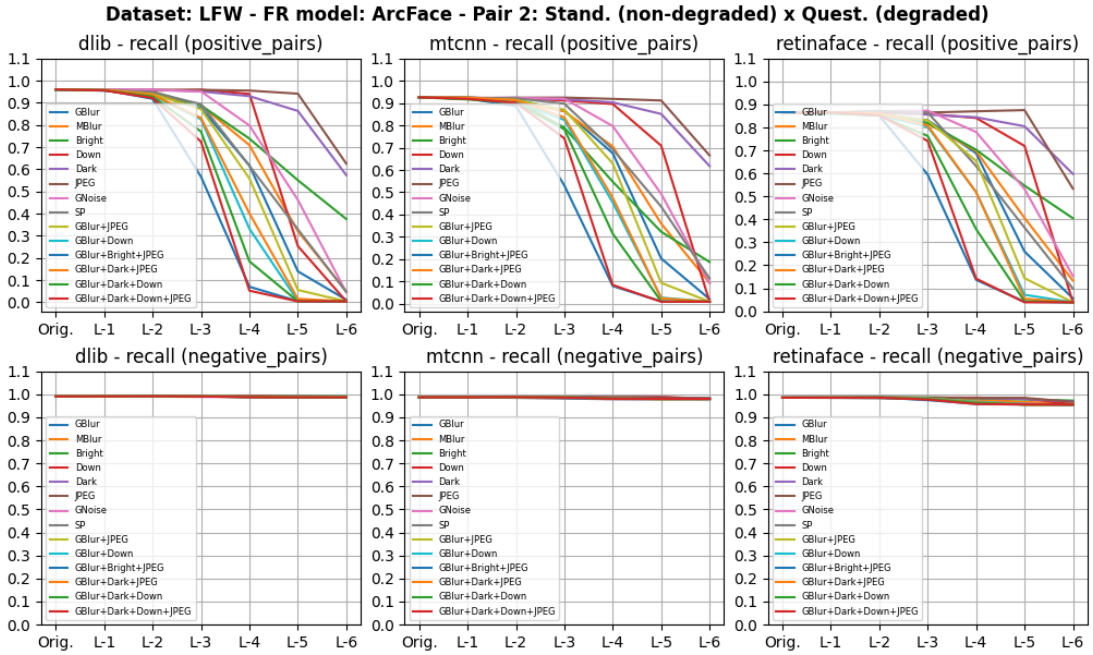


Figura I.9: Dataset LFW - Par 2 - Métrica *recall* do algoritmo ArcFace

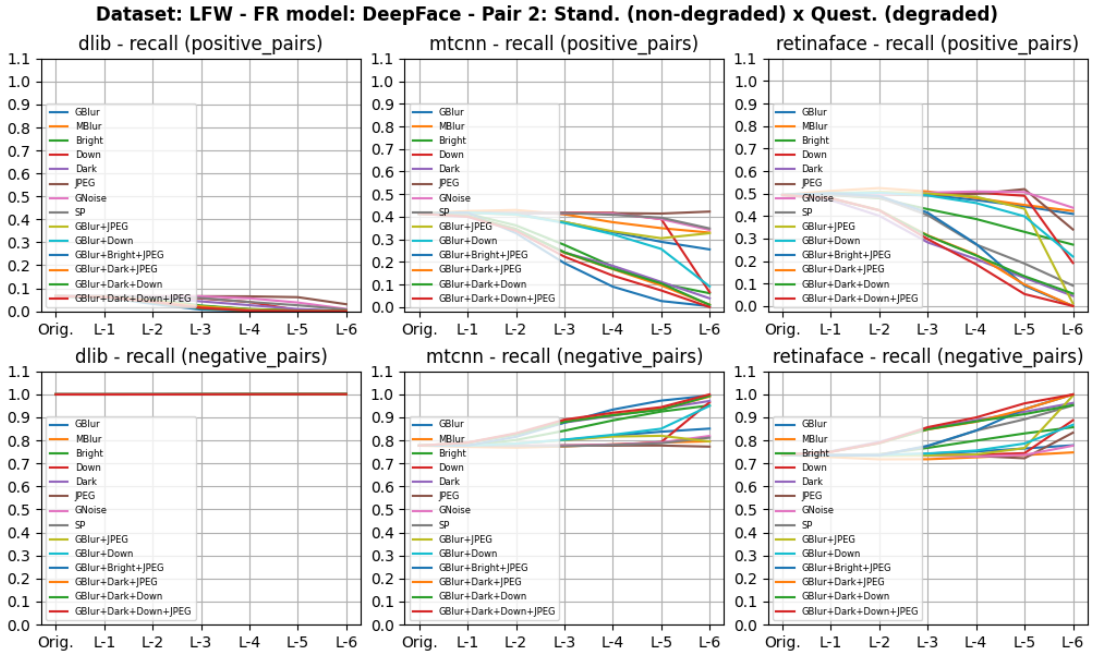


Figura I.10: Dataset LFW - Par 2 - Métrica *recall* do algoritmo DeepFace

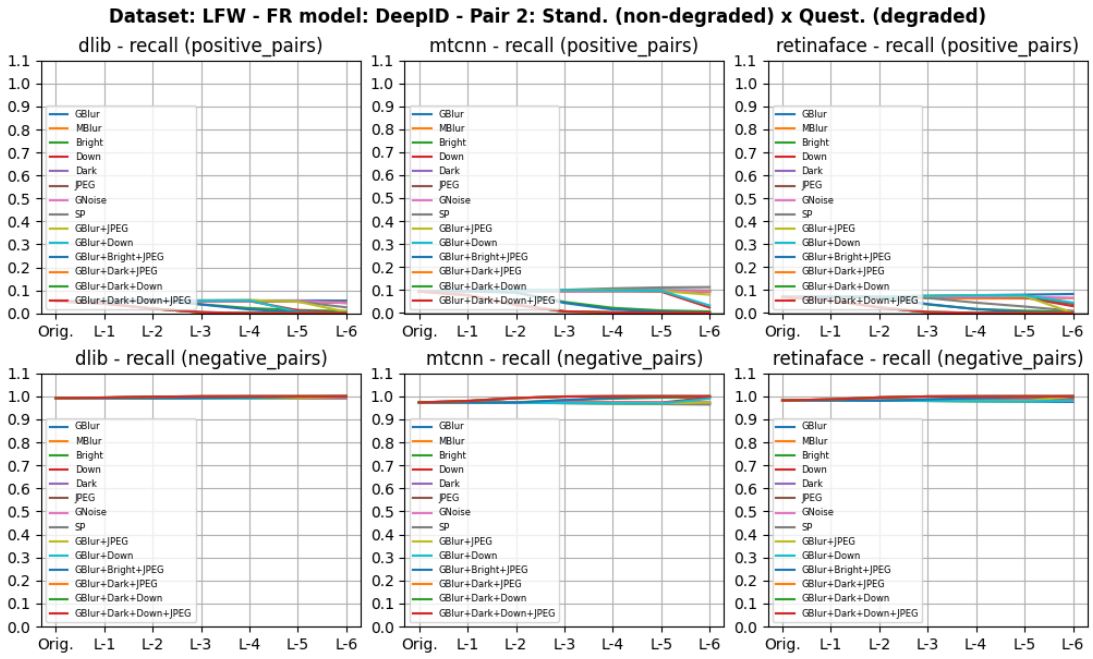


Figura I.11: Dataset LFW - Par 2 - Métrica *recall* do algoritmo DeepID

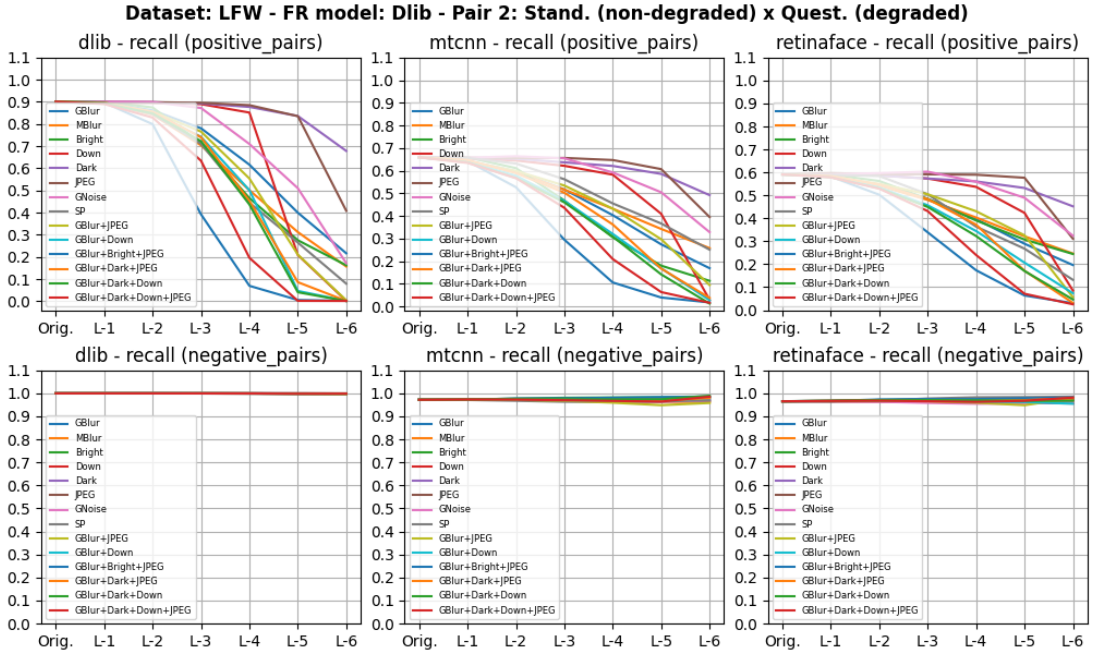


Figura I.12: Dataset LFW - Par 2 - Métrica *recall* do algoritmo Dlib

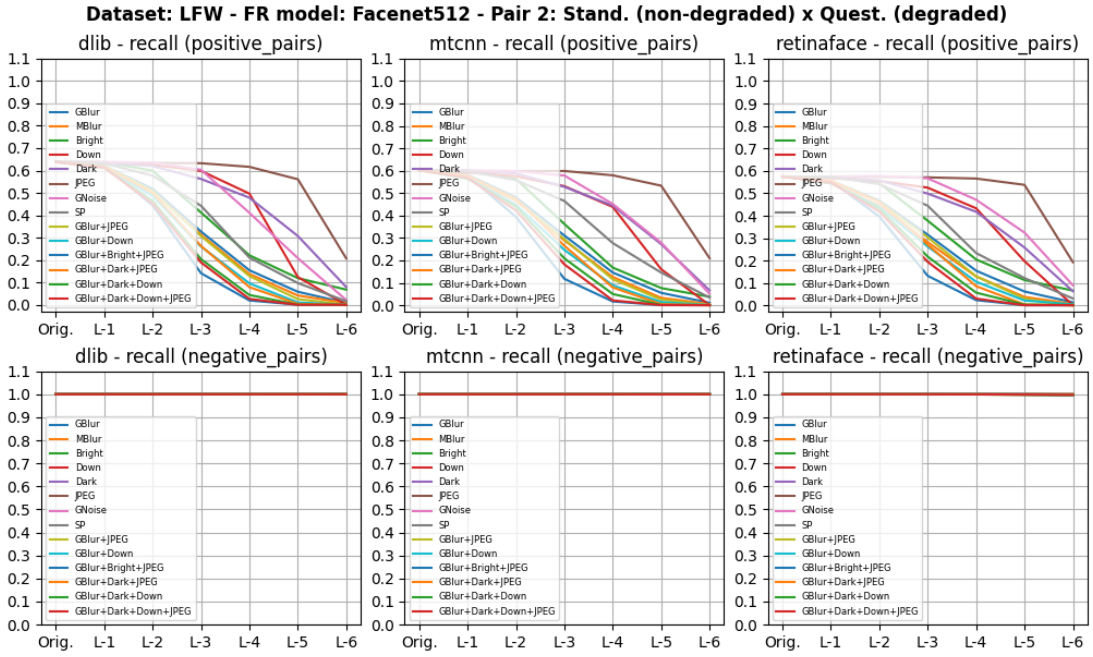


Figura I.13: Dataset LFW - Par 2 - Métrica *recall* do algoritmo FaceNet512

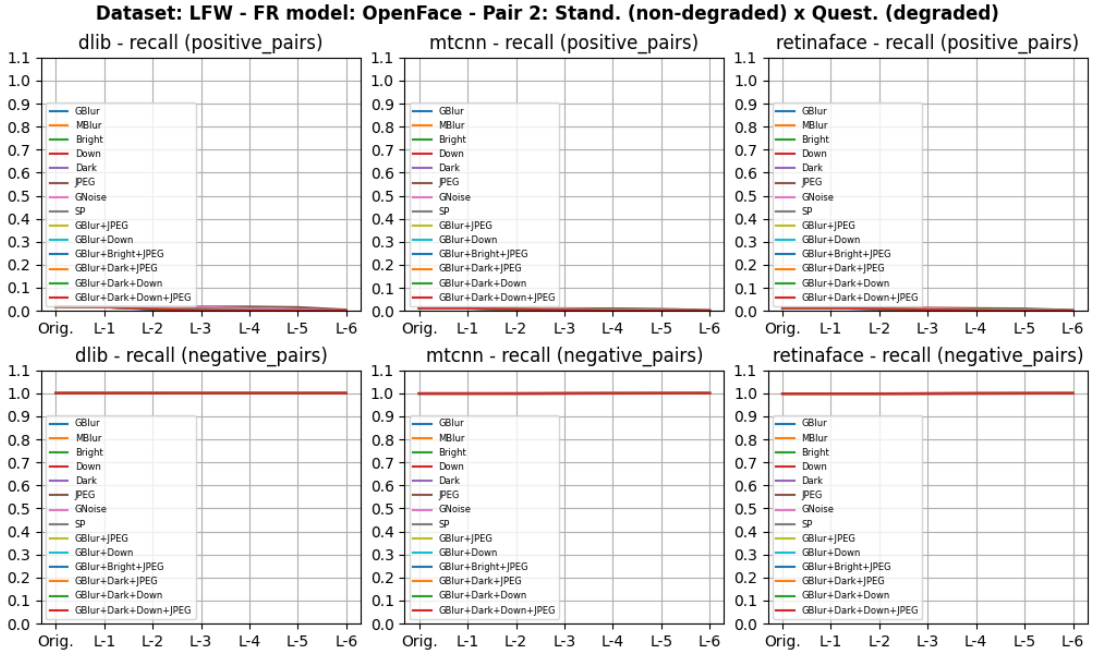


Figura I.14: Dataset LFW - Par 2 - Métrica *recall* do algoritmo OpenFace

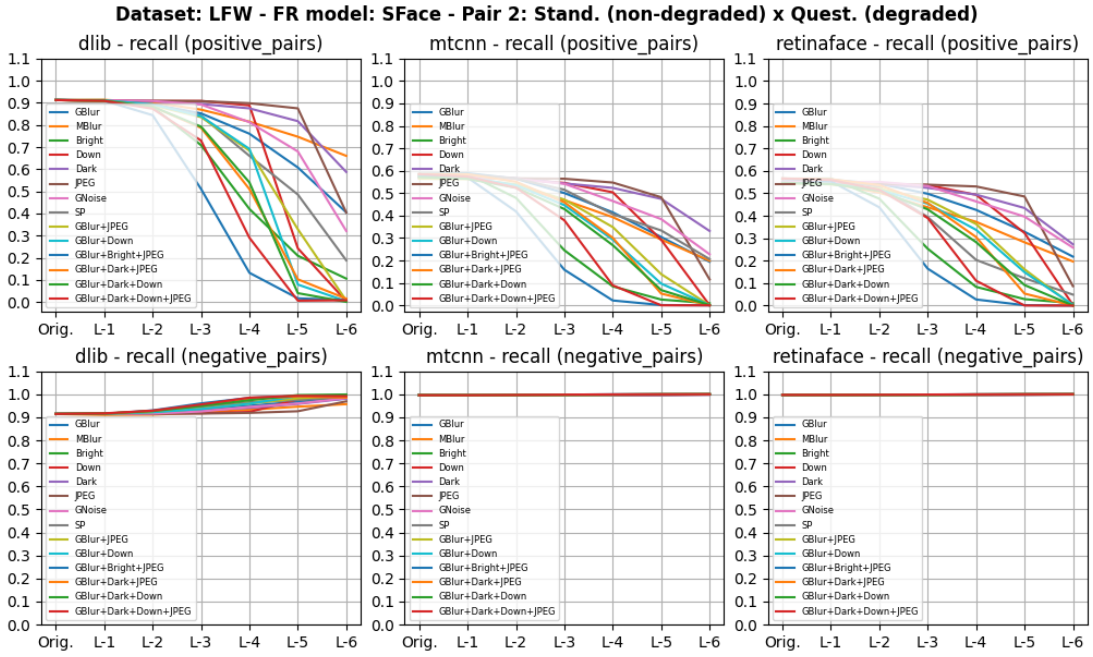


Figura I.15: Dataset LFW - Par 2 - Métrica *recall* do algoritmo SFace

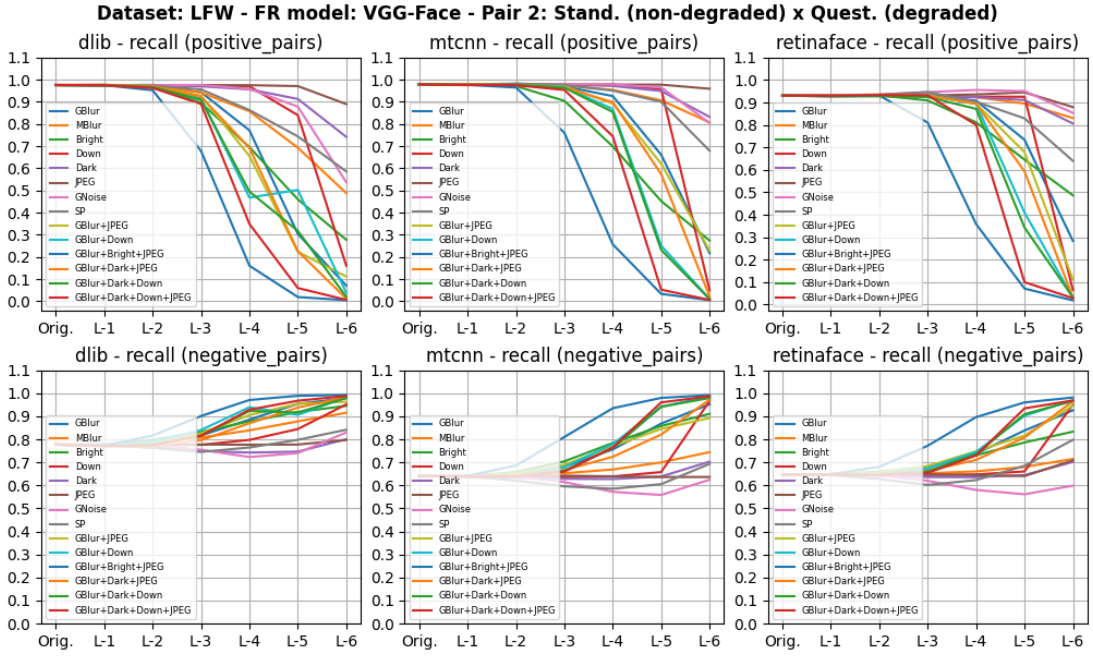


Figura I.16: Dataset LFW - Par 2 - Métrica *recall* do algoritmo VGG

Dataset LFW: Par 3 - Imagem padrão (degradada) x Imagem questionada (degradada)

As imagens deste par são as imagens compreendidas da Figura I.17 até a Figura I.24

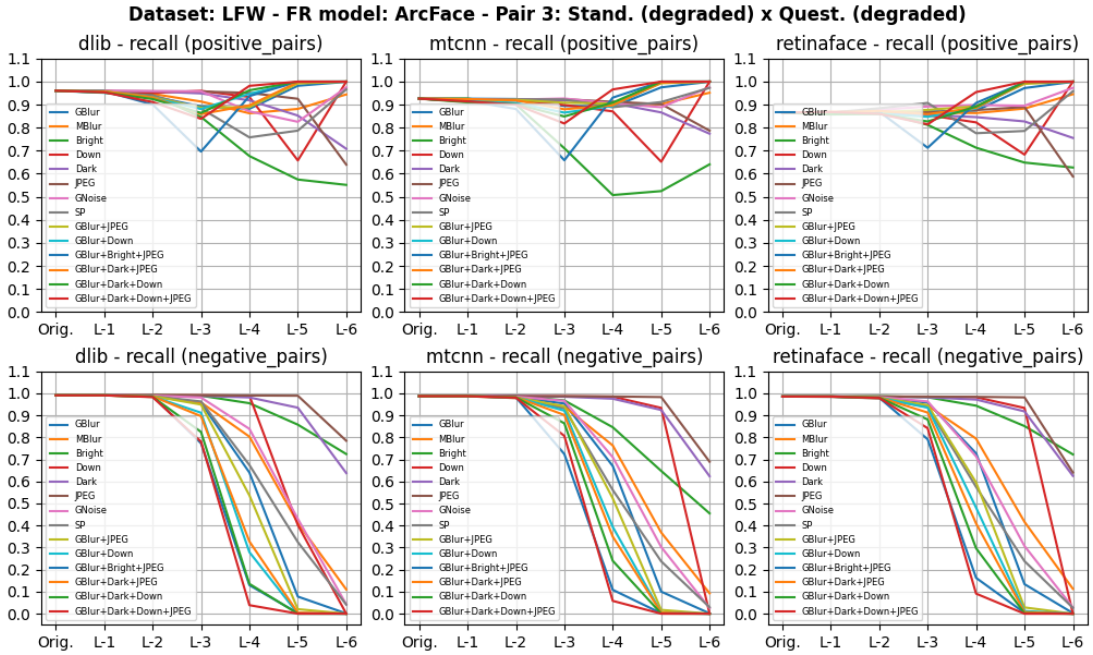


Figura I.17: Dataset LFW - Par 3 - Métrica *recall* do algoritmo ArcFace

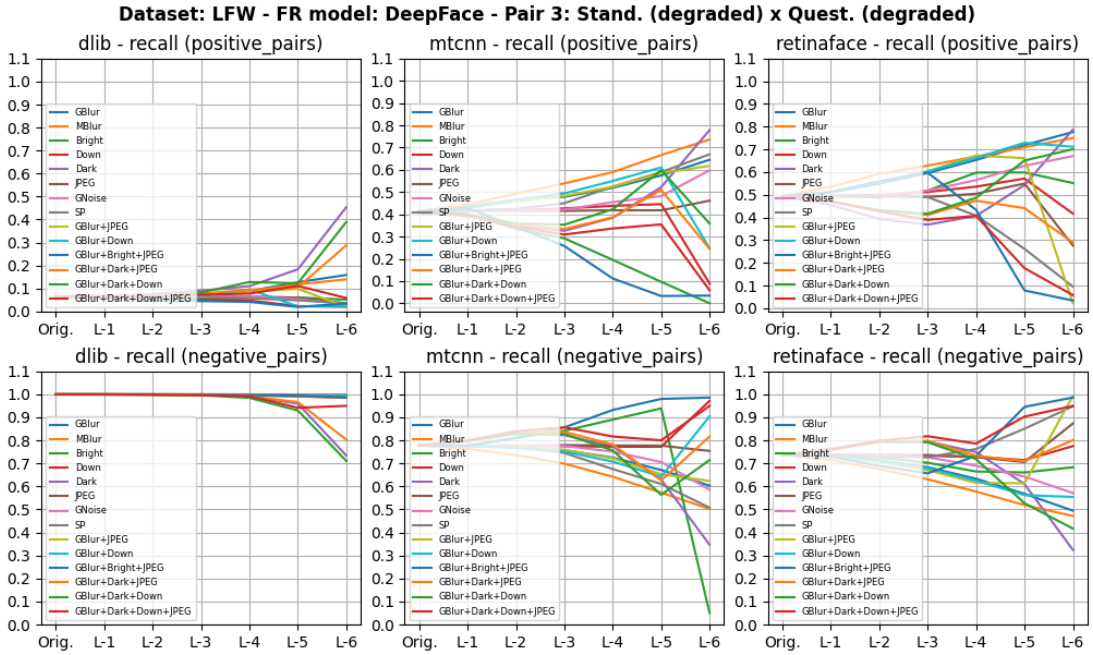


Figura I.18: Dataset LFW - Par 3 - Métrica *recall* do algoritmo DeepFace

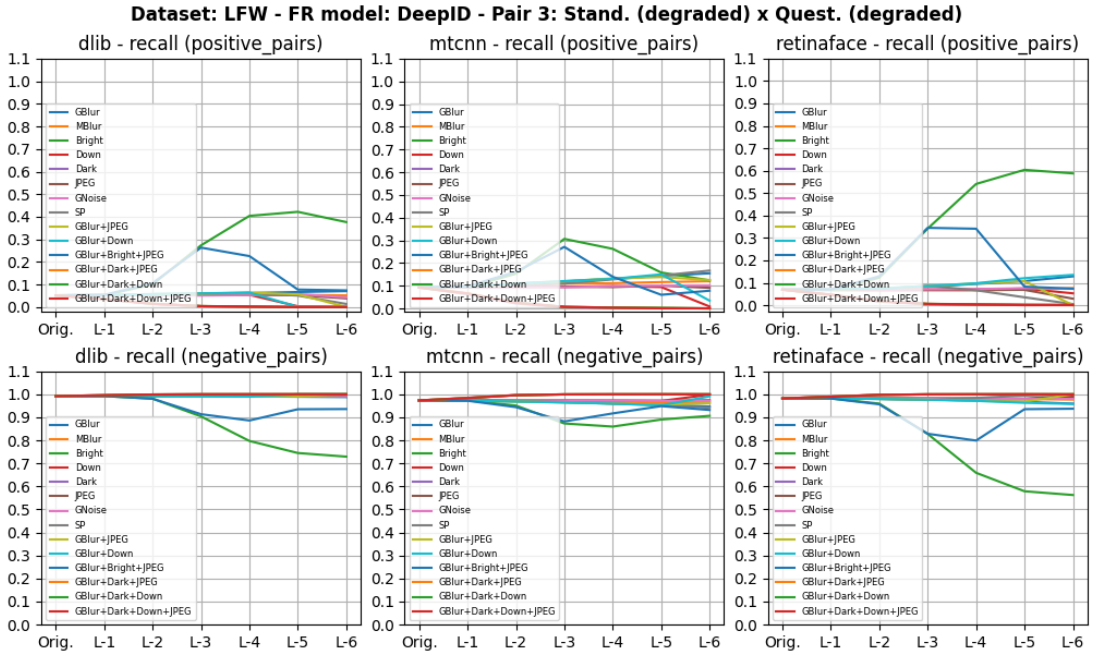


Figura I.19: Dataset LFW - Par 3 - Métrica *recall* do algoritmo DeepID

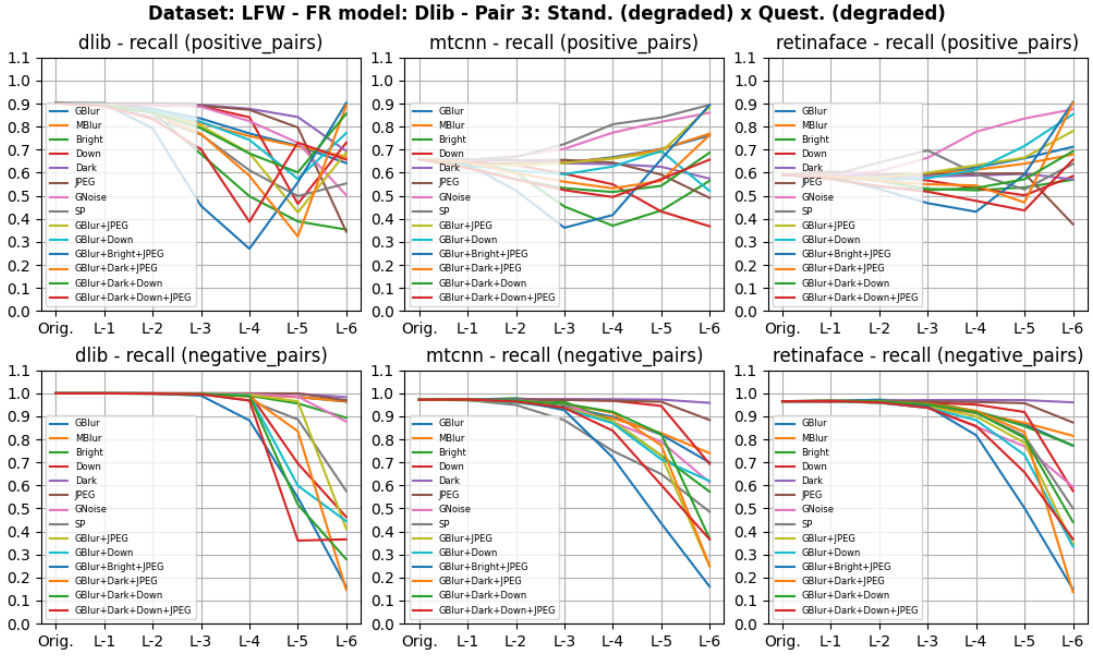


Figura I.20: Dataset LFW - Par 3 - Métrica *recall* do algoritmo Dlib

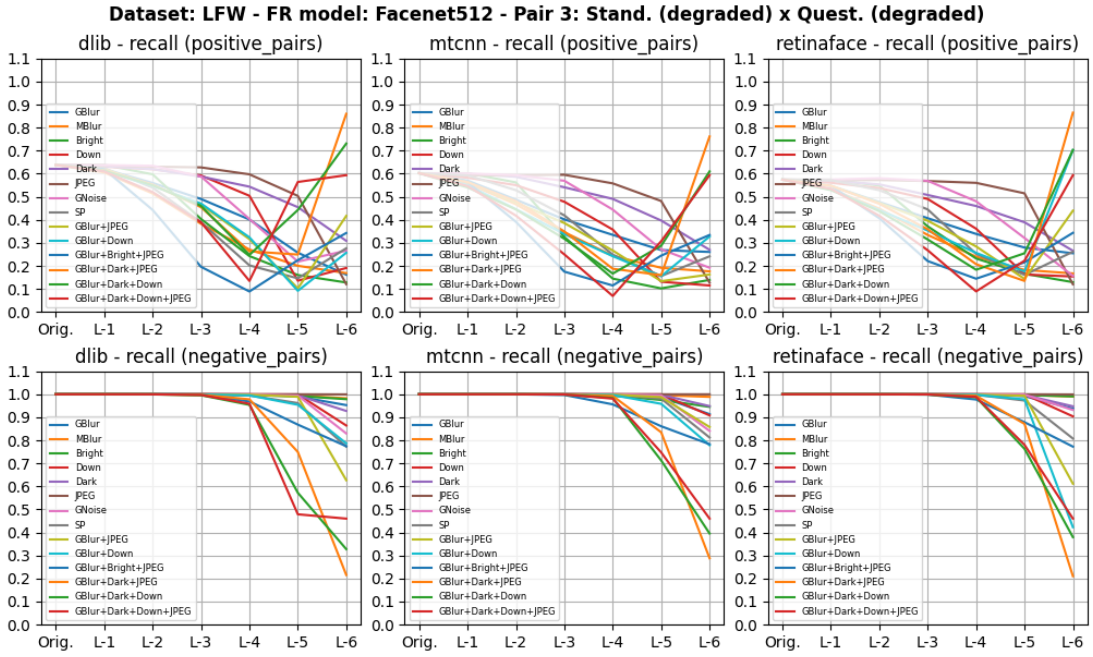


Figura I.21: Dataset LFW - Par 3 - Métrica *recall* do algoritmo FaceNet512

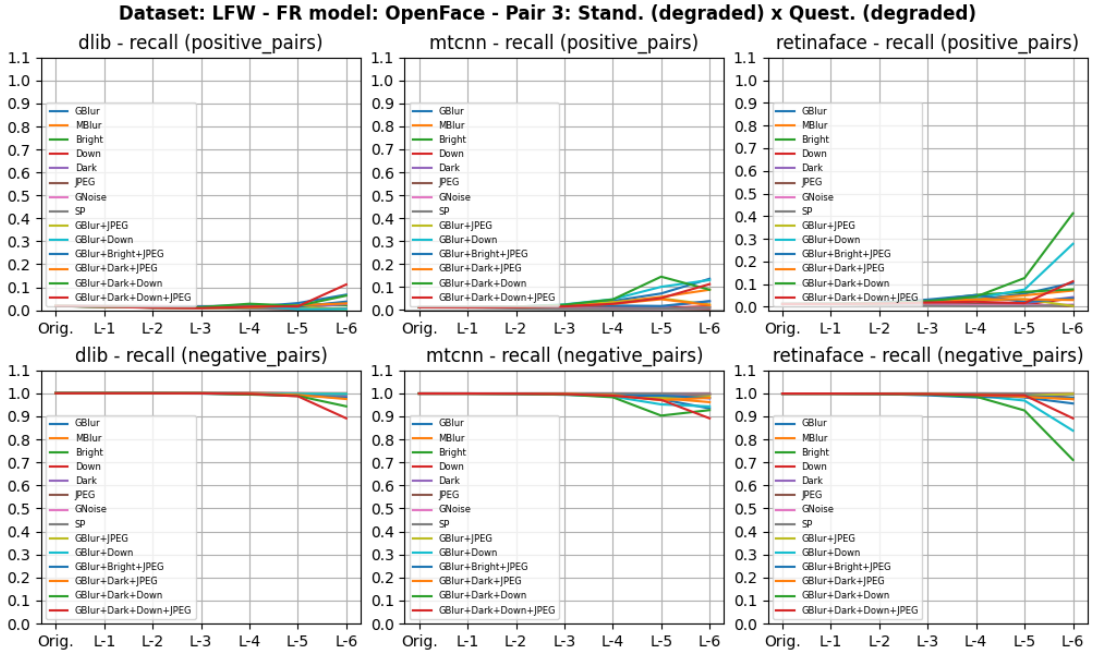


Figura I.22: Dataset LFW - Par 3 - Métrica *recall* do algoritmo OpenFace

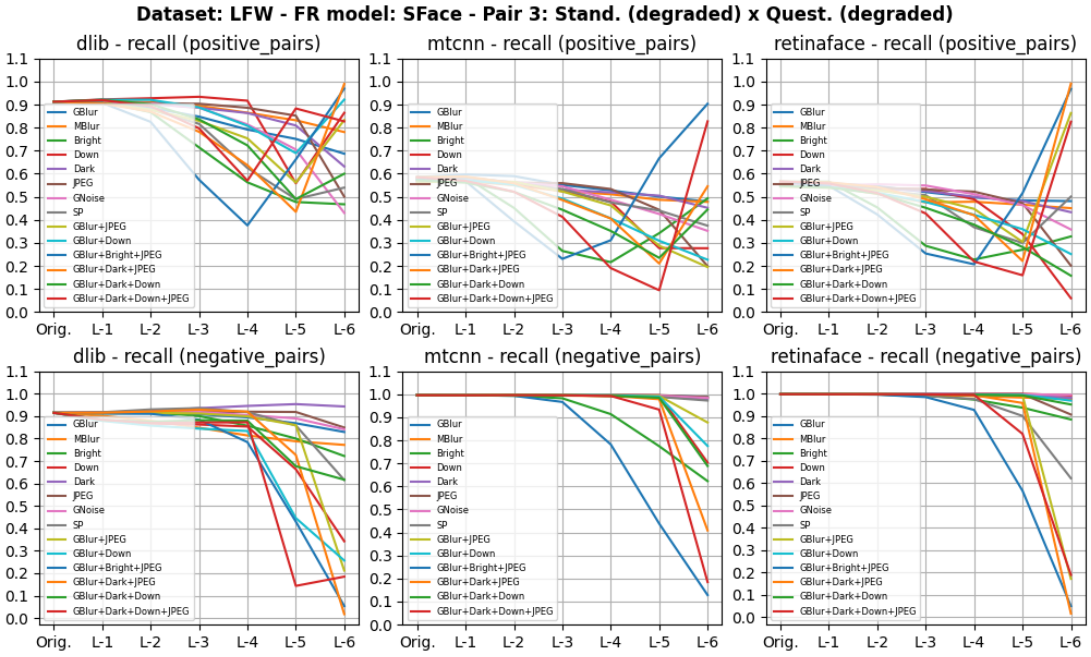


Figura I.23: Dataset LFW - Par 3 - Métrica *recall* do algoritmo SFace

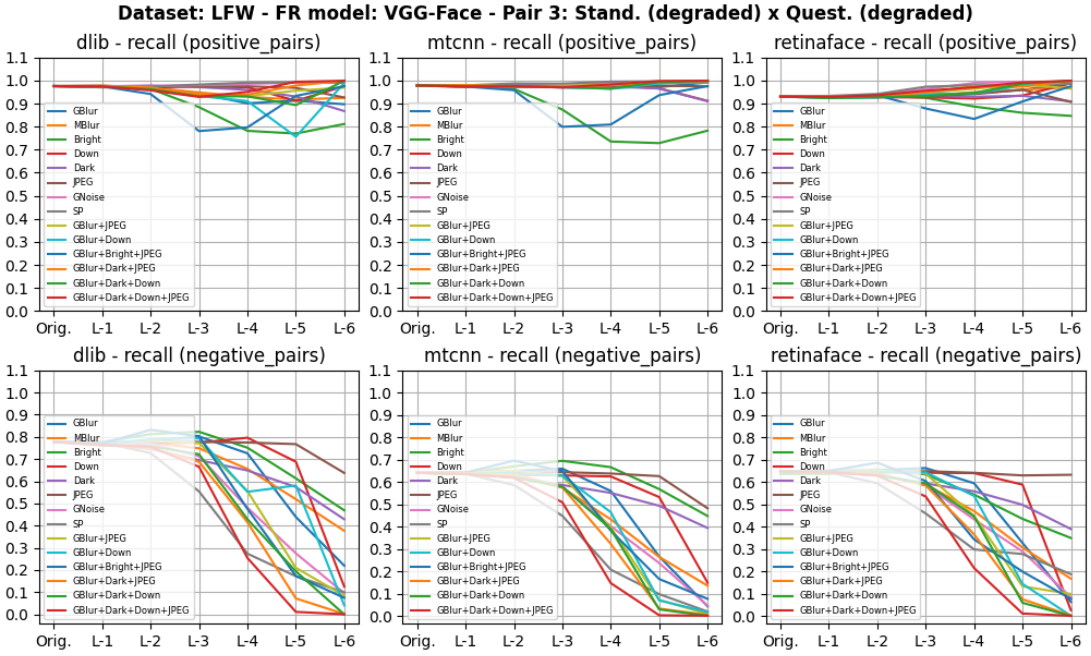


Figura I.24: Dataset LFW - Par 3 - Métrica *recall* do algoritmo VGG

Dataset FEI: Par 1 - Imagem padrão (não degradada) x Cópia da imagem padrão (degradada)

As imagens deste par são as imagens compreendidas da Figura I.25 até a Figura I.32

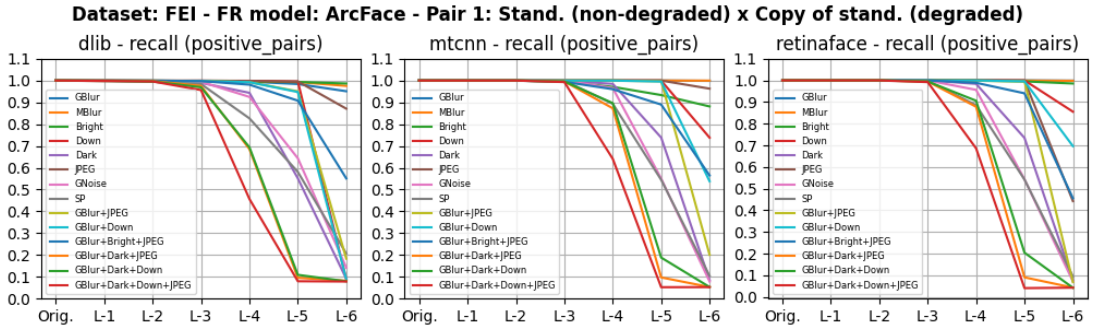


Figura I.25: Dataset FEI - Par 1 - Métrica *recall* do algoritmo ArcFace

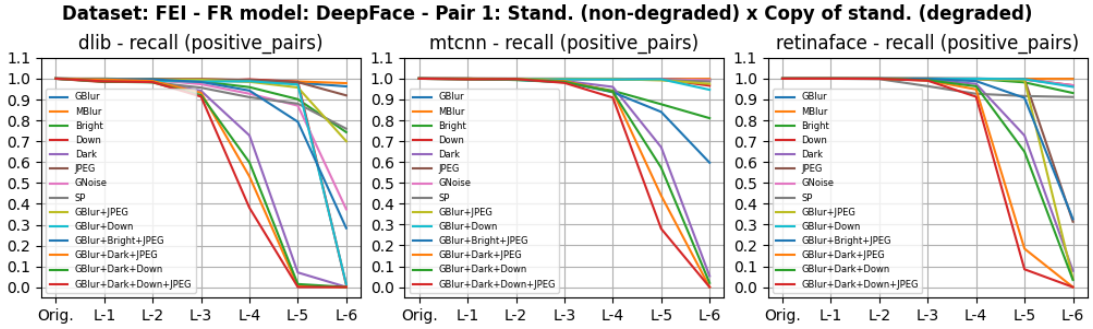


Figura I.26: Dataset FEI - Par 1 - Métrica *recall* do algoritmo DeepFace

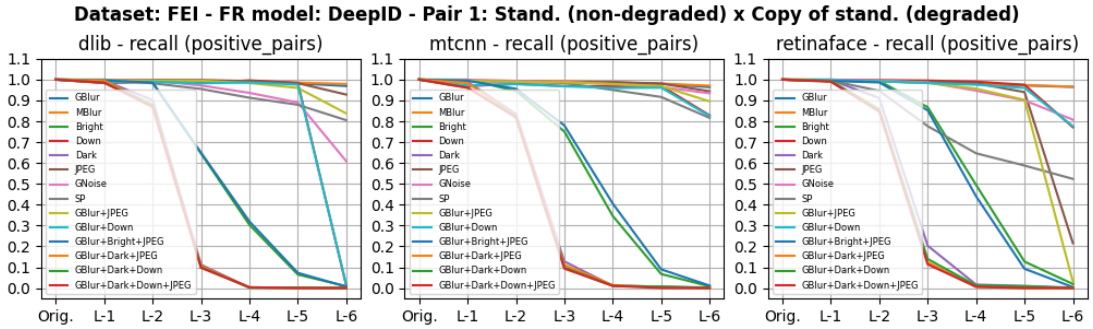


Figura I.27: Dataset FEI - Par 1 - Métrica *recall* do algoritmo DeepID

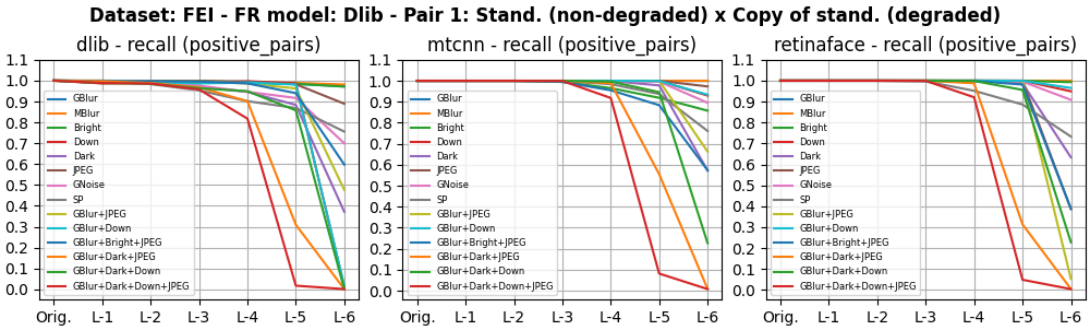


Figura I.28: Dataset FEI - Par 1 - Métrica *recall* do algoritmo Dlib

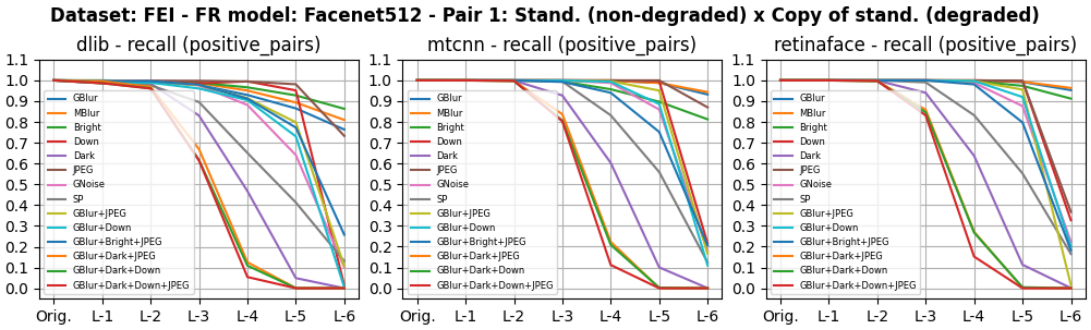


Figura I.29: Dataset FEI - Par 1 - Métrica *recall* do algoritmo FaceNet512

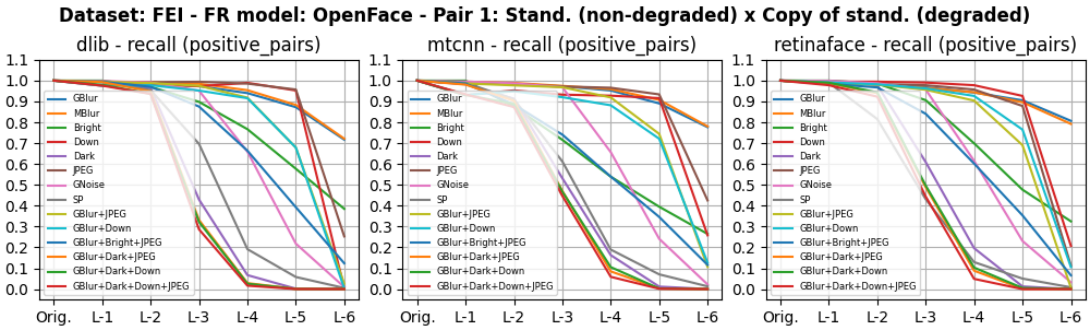


Figura I.30: Dataset FEI - Par 1 - Métrica *recall* do algoritmo OpenFace

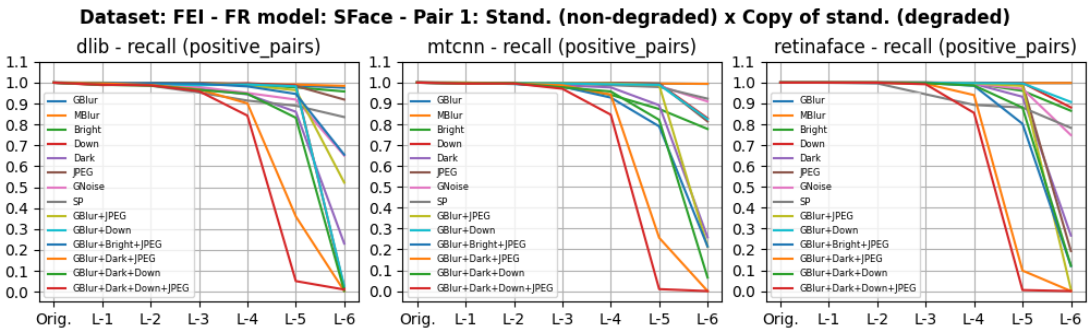


Figura I.31: Dataset FEI - Par 1 - Métrica *recall* do algoritmo SFace

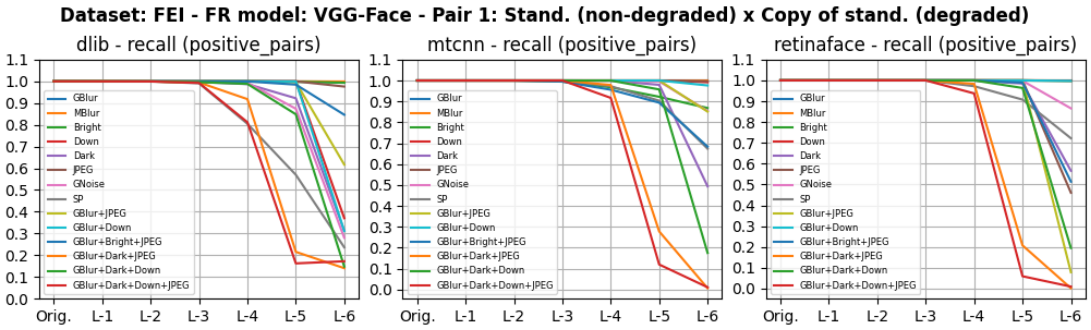


Figura I.32: Dataset FEI - Par 1 - Métrica *recall* do algoritmo VGG

Dataset FEI: Par 2 - Imagem padrão (não degradada) x Imagem questionada (degradada)

As imagens deste par são as imagens compreendidas da Figura I.33 até a Figura I.40

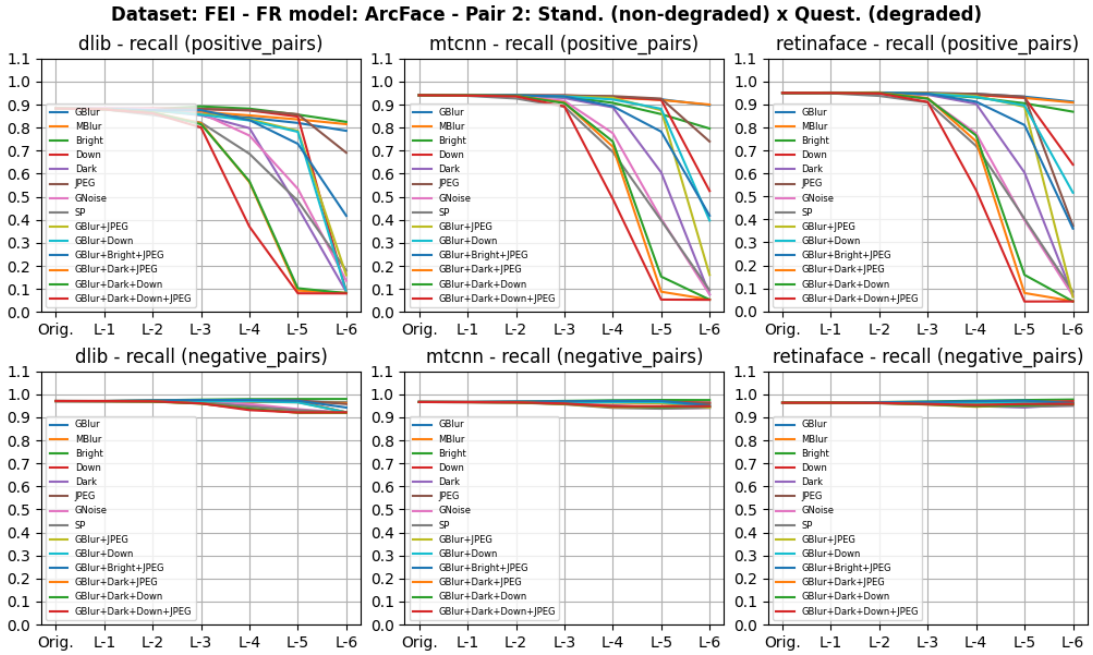


Figura I.33: Dataset FEI - Par 2 - Métrica *recall* do algoritmo ArcFace

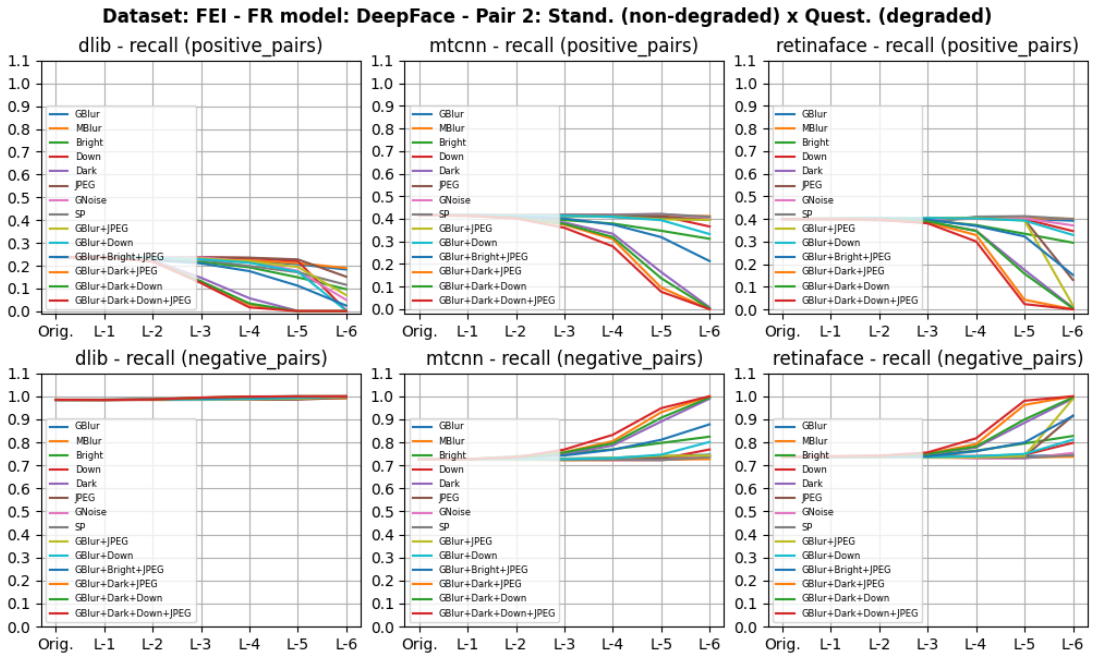


Figura I.34: Dataset FEI - Par 2 - Métrica *recall* do algoritmo DeepFace

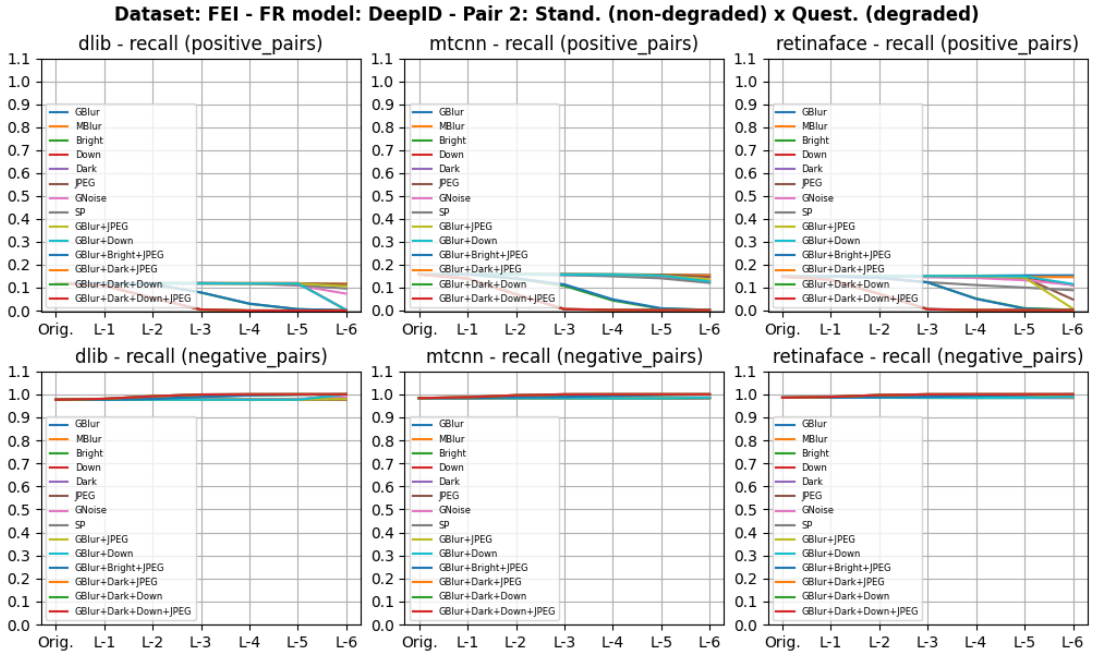


Figura I.35: Dataset FEI - Par 2 - Métrica *recall* do algoritmo DeepID

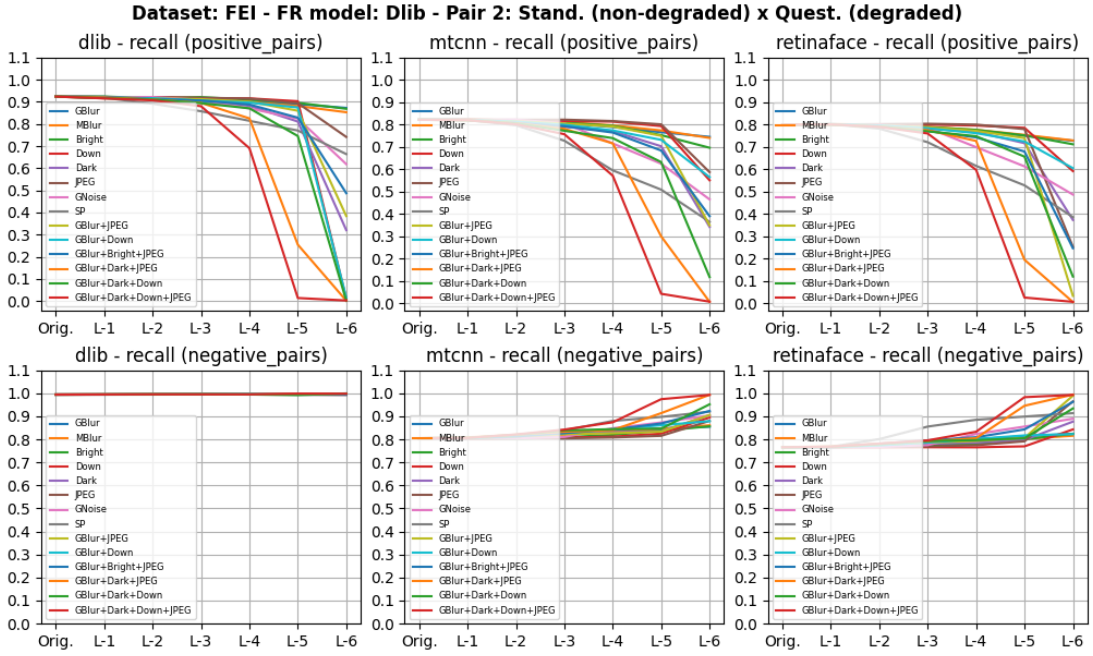


Figura I.36: Dataset FEI - Par 2 - Métrica *recall* do algoritmo Dlib

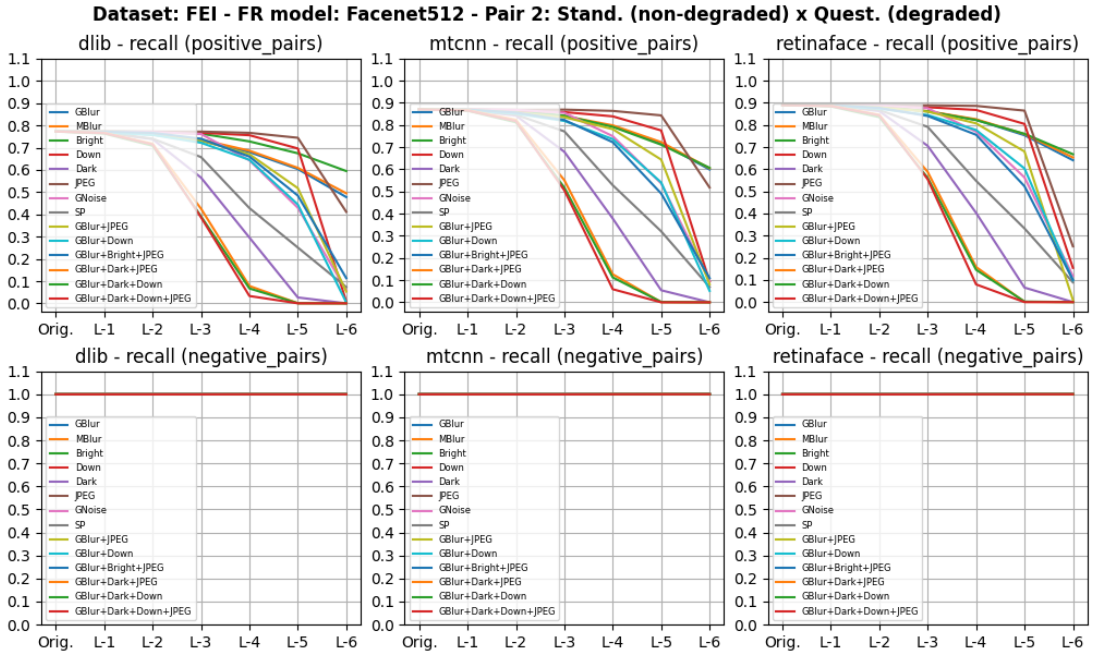


Figura I.37: Dataset FEI - Par 2 - Métrica *recall* do algoritmo FaceNet512

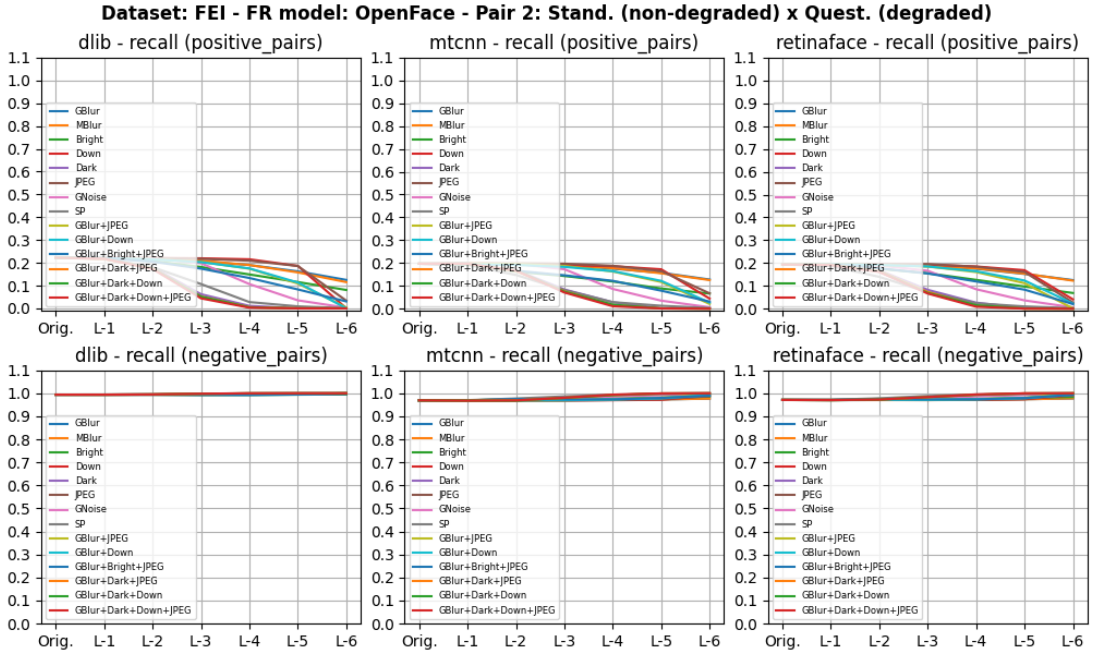


Figura I.38: Dataset FEI - Par 2 - Métrica *recall* do algoritmo OpenFace

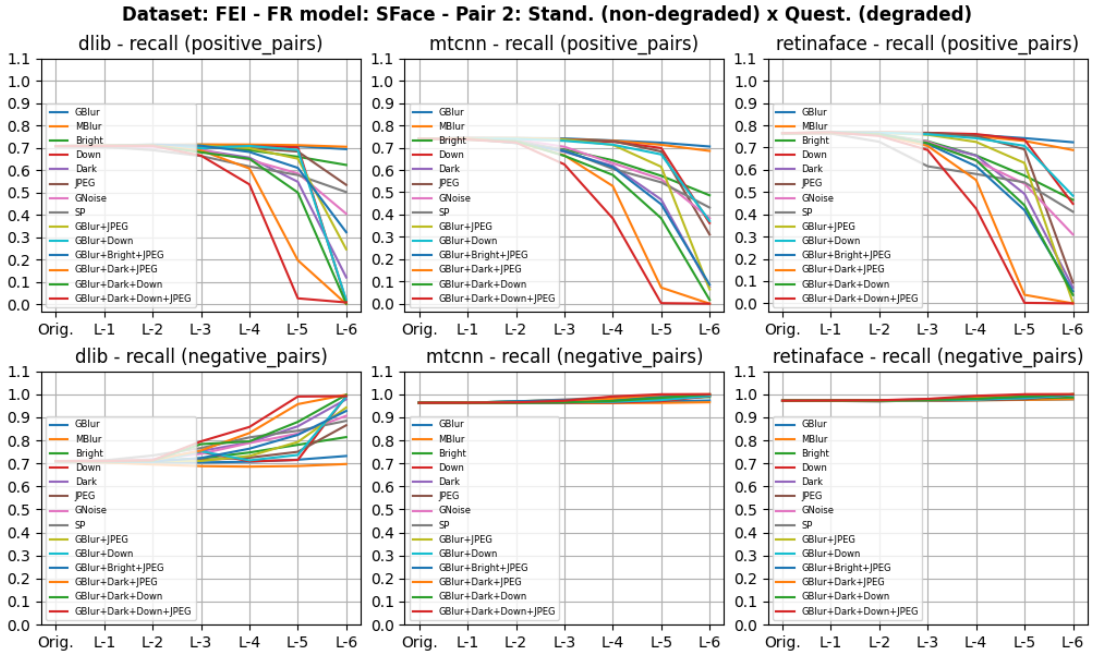


Figura I.39: Dataset FEI - Par 2 - Métrica *recall* do algoritmo SFace

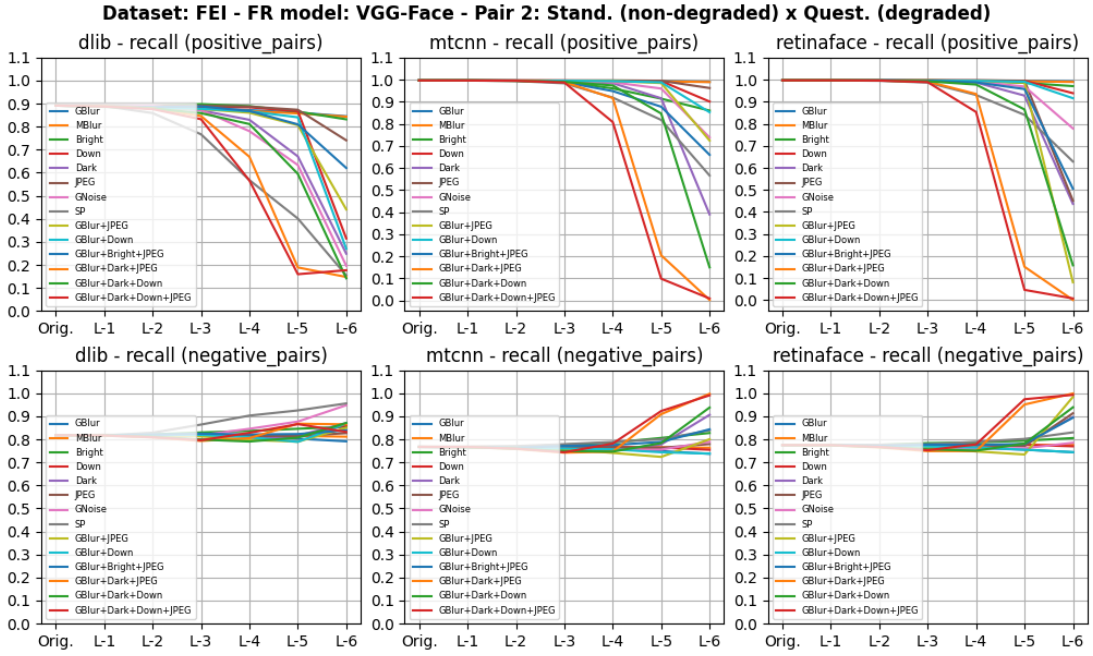


Figura I.40: Dataset FEI - Par 2 - Métrica *recall* do algoritmo VGG

Dataset FEI: Par 3 - Imagem padrão (degradada) x Imagem questionada (degradada)

As imagens deste par são as imagens compreendidas da Figura I.41 até a Figura I.48

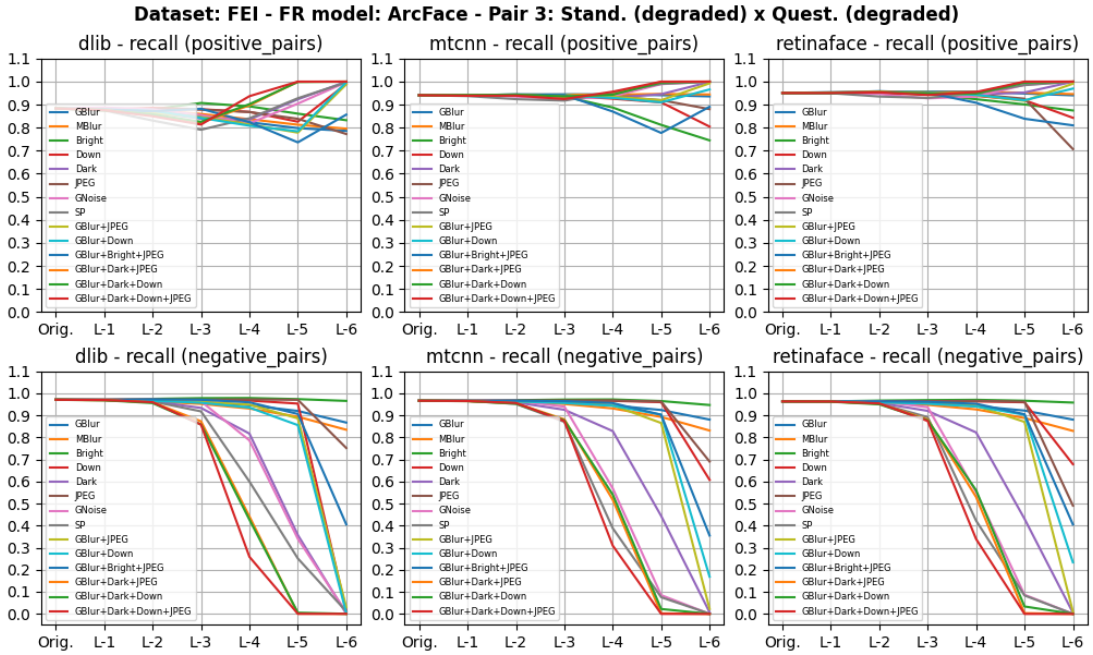


Figura I.41: Dataset FEI - Par 3 - Métrica *recall* do algoritmo ArcFace

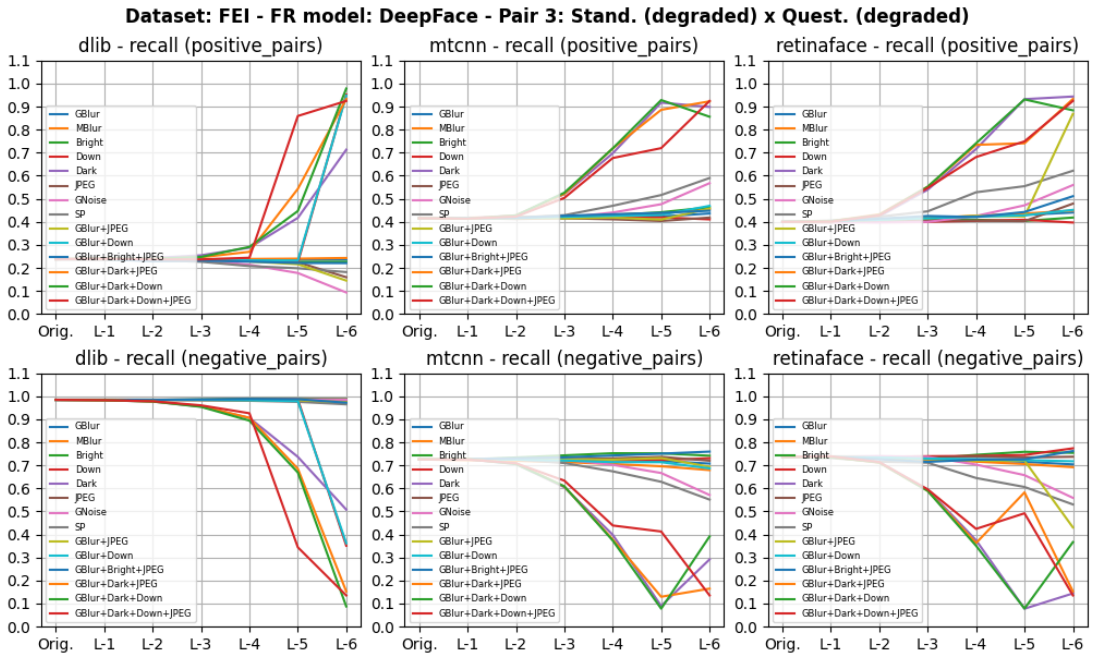


Figura I.42: Dataset FEI - Par 3 - Métrica *recall* do algoritmo DeepFace

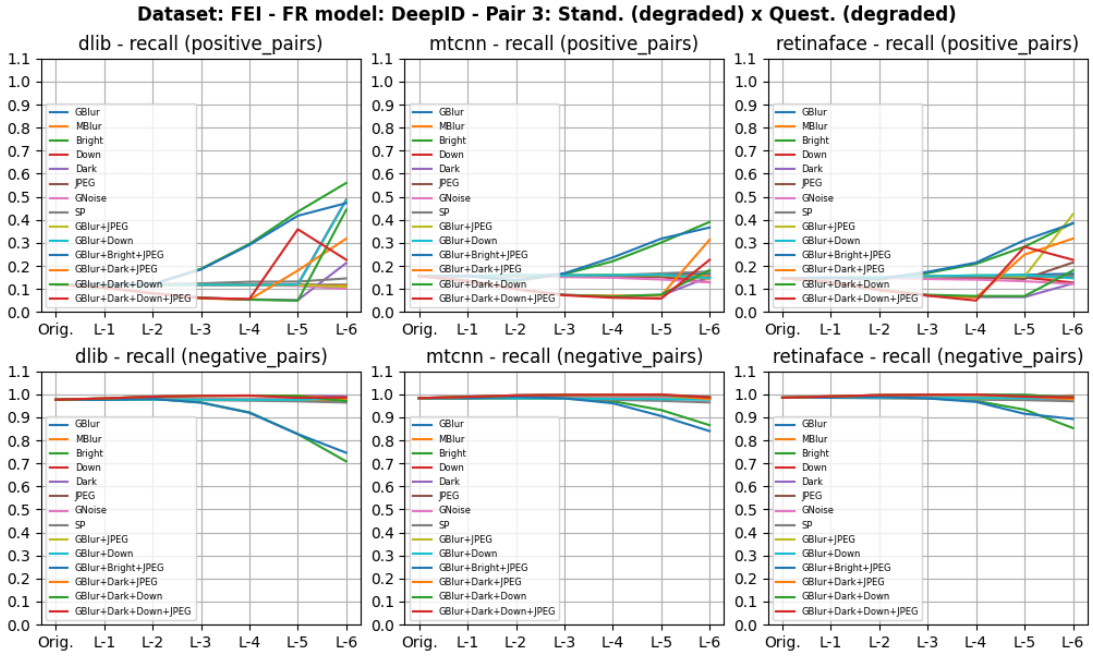


Figura I.43: Dataset FEI - Par 3 - Métrica *recall* do algoritmo DeepID

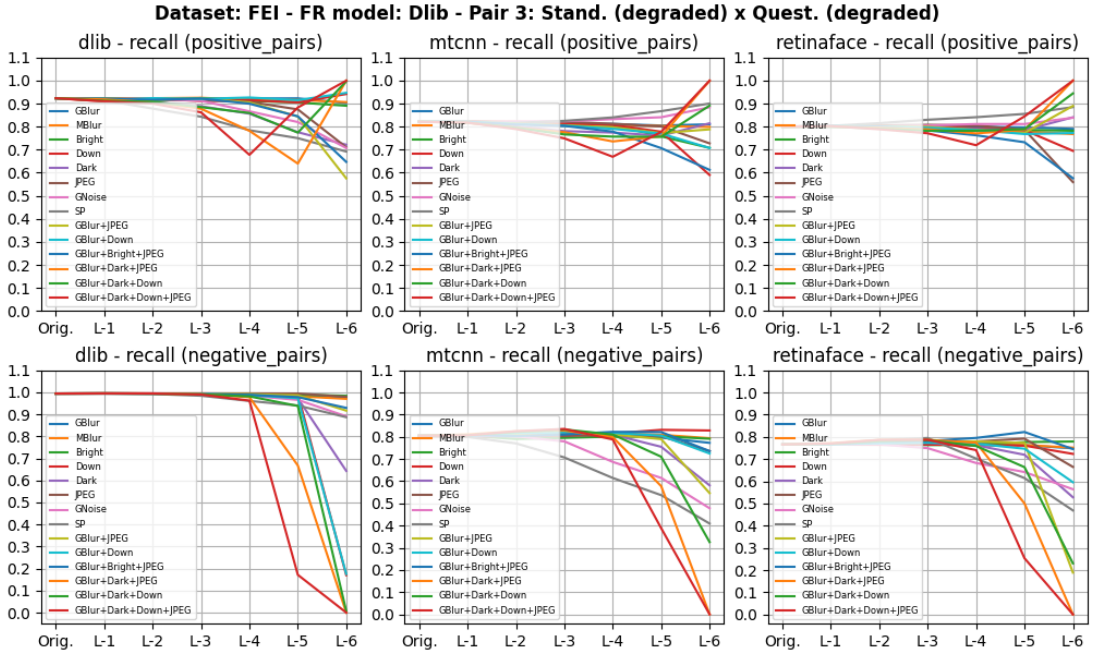


Figura I.44: Dataset FEI - Par 3 - Métrica *recall* do algoritmo Dlib

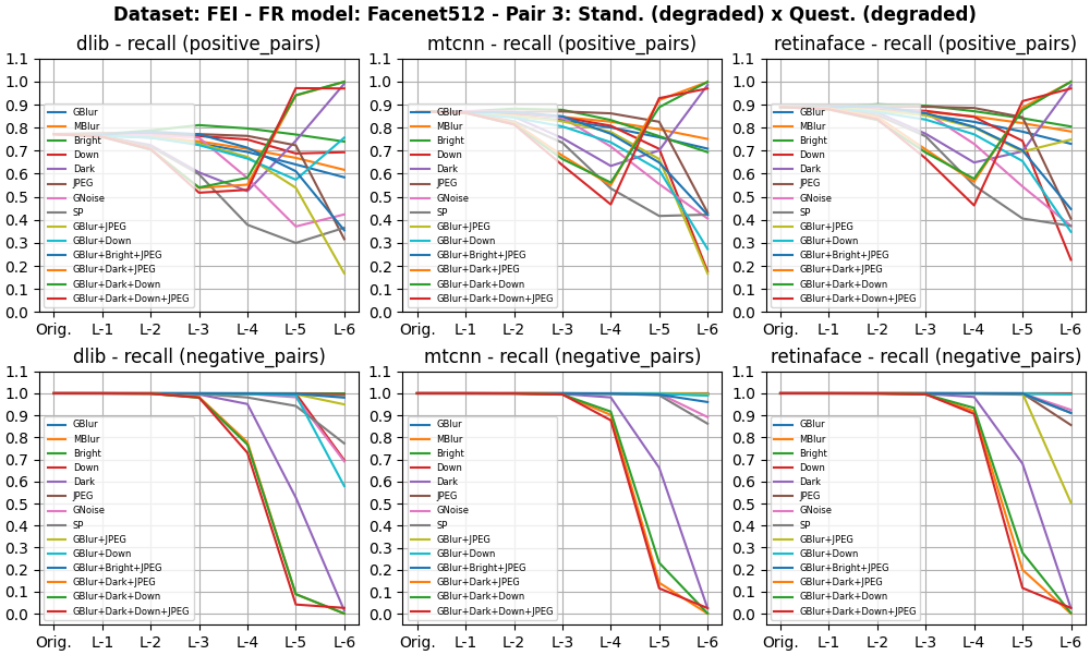


Figura I.45: Dataset FEI - Par 3 - Métrica *recall* do algoritmo FaceNet512

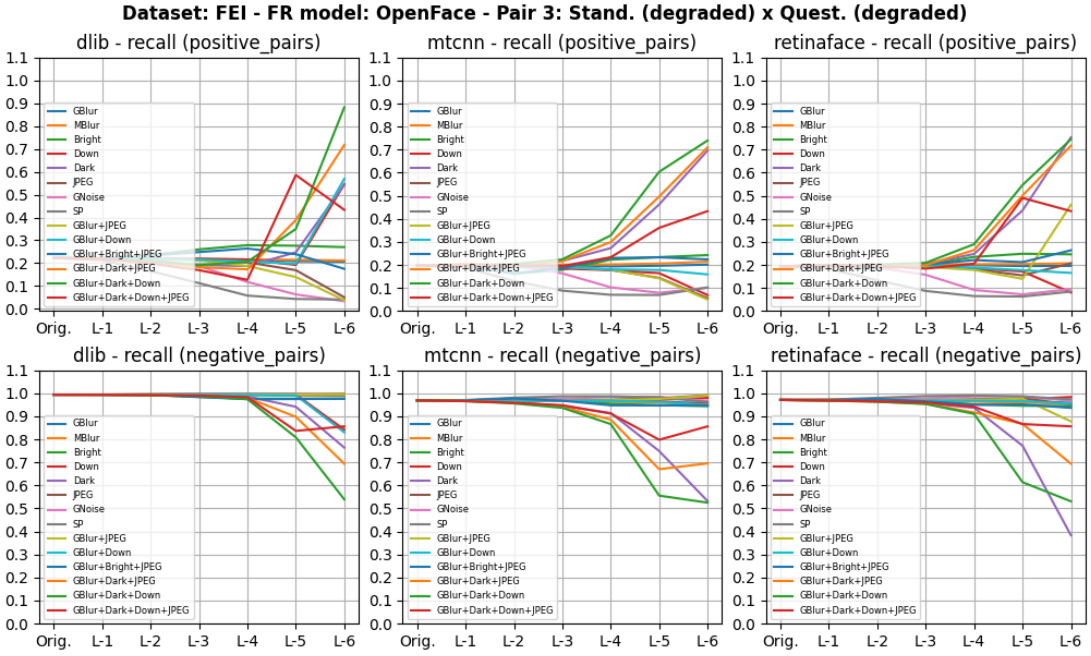


Figura I.46: Dataset FEI - Par 3 - Métrica *recall* do algoritmo OpenFace

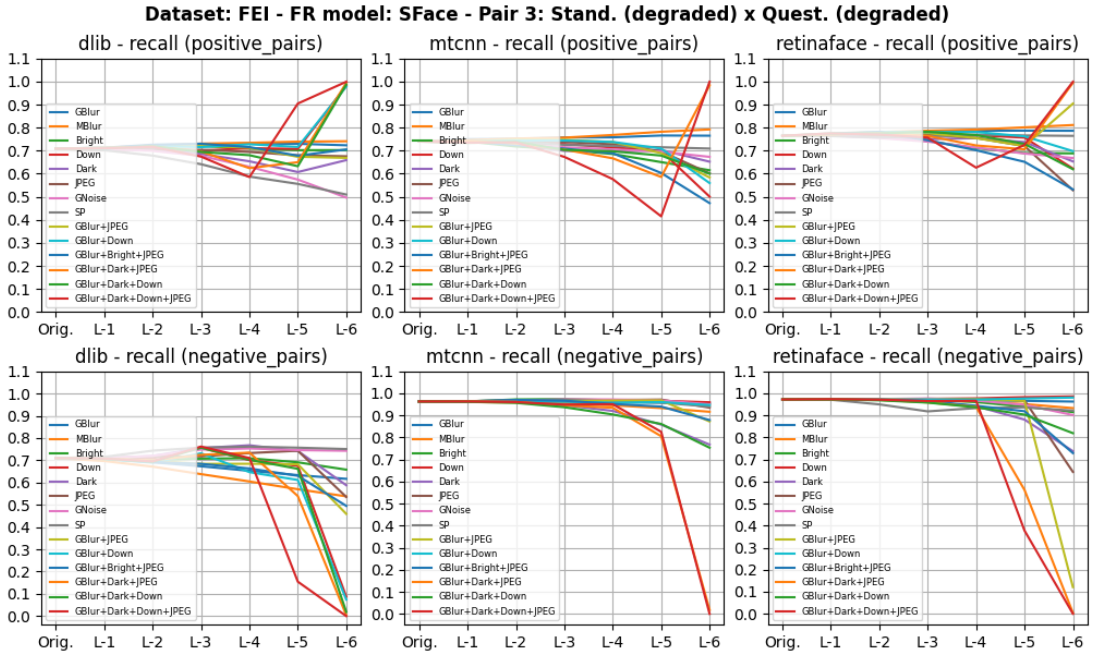


Figura I.47: Dataset FEI - Par 3 - Métrica *recall* do algoritmo SFace

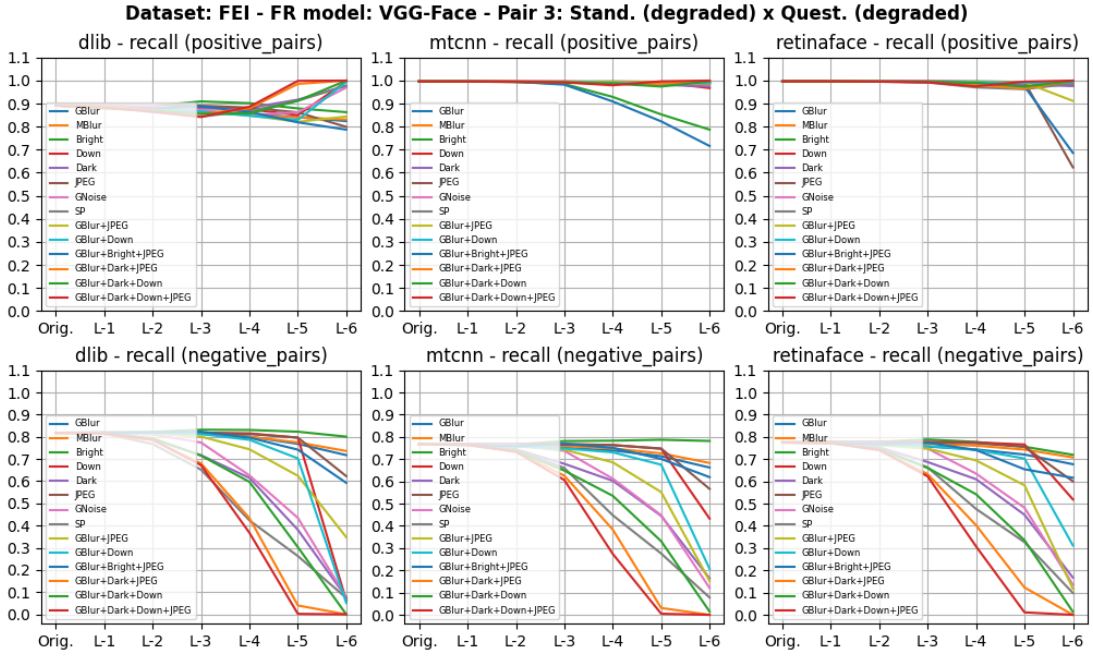


Figura I.48: Dataset FEI - Par 3 - Métrica *recall* do algoritmo VGG

Dataset GUFD: Par 1 - Imagem padrão (não degradada) x Cópia da imagem padrão (degradada)

As imagens deste par são as imagens compreendidas da Figura **??** até a Figura I.56

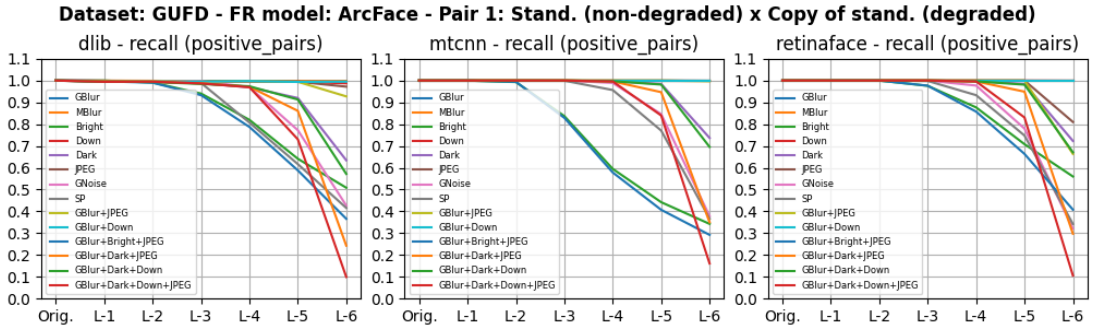


Figura I.49: Dataset GUFD - Par 1 - Métrica *recall* do algoritmo ArcFace

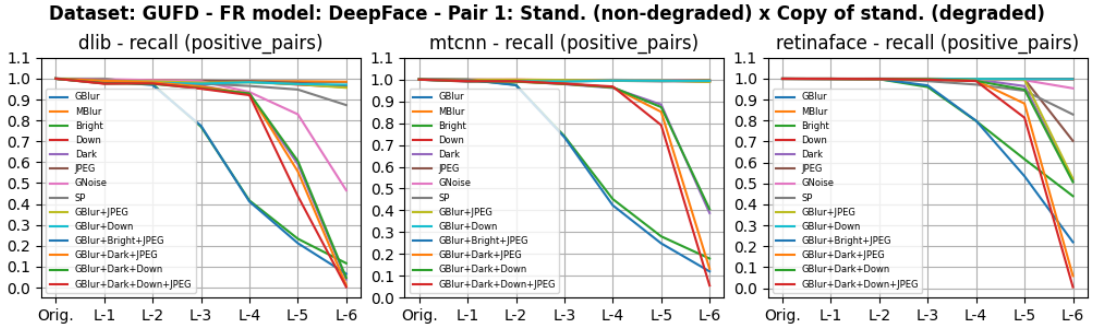


Figura I.50: Dataset GUFD - Par 1 - Métrica *recall* do algoritmo DeepFace

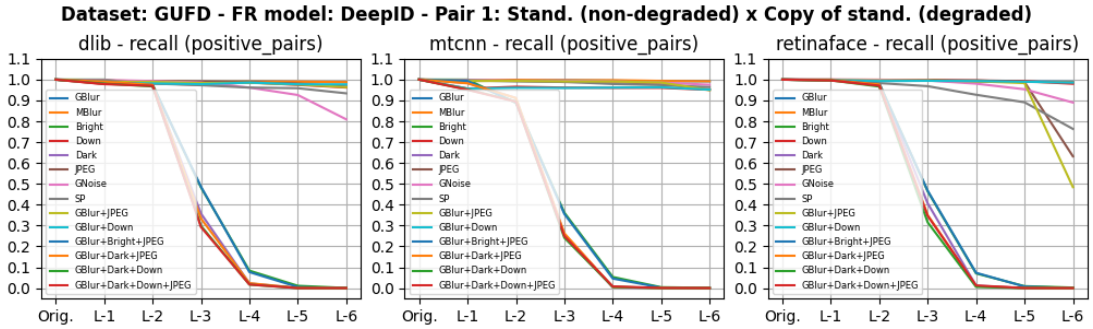


Figura I.51: Dataset GUFD - Par 1 - Métrica *recall* do algoritmo DeepID

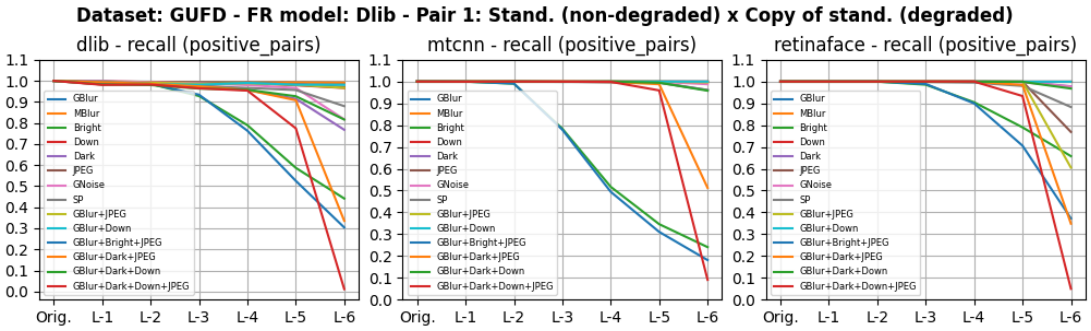


Figura I.52: Dataset GUFD - Par 1 - Métrica *recall* do algoritmo Dlib

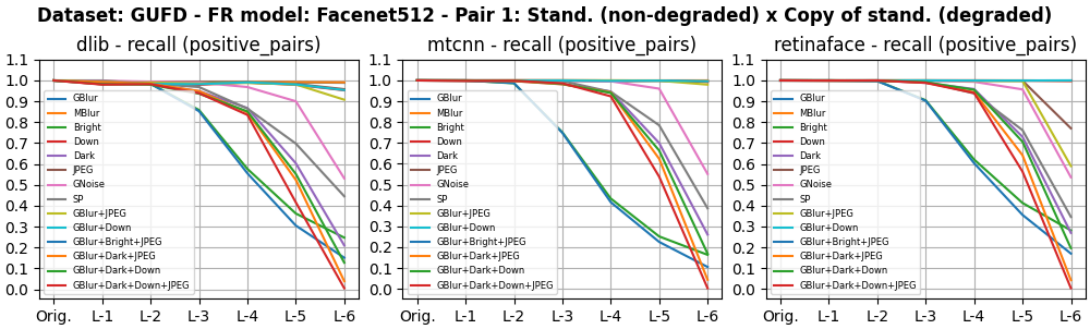


Figura I.53: Dataset GUFD - Par 1 - Métrica *recall* do algoritmo FaceNet512

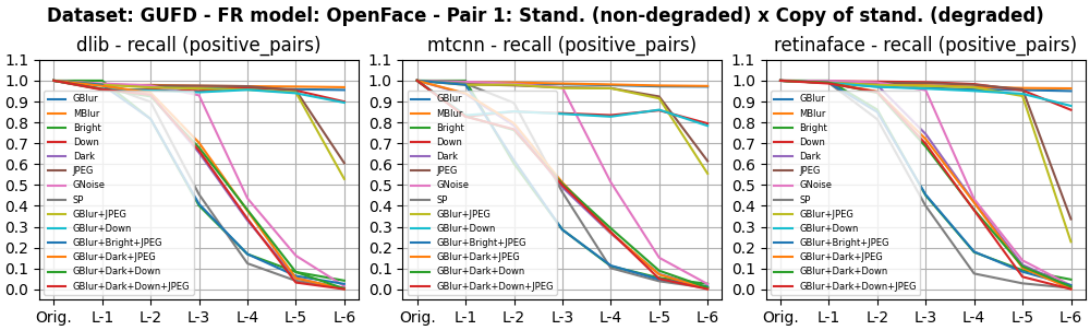


Figura I.54: Dataset GUFD - Par 1 - Métrica *recall* do algoritmo OpenFace

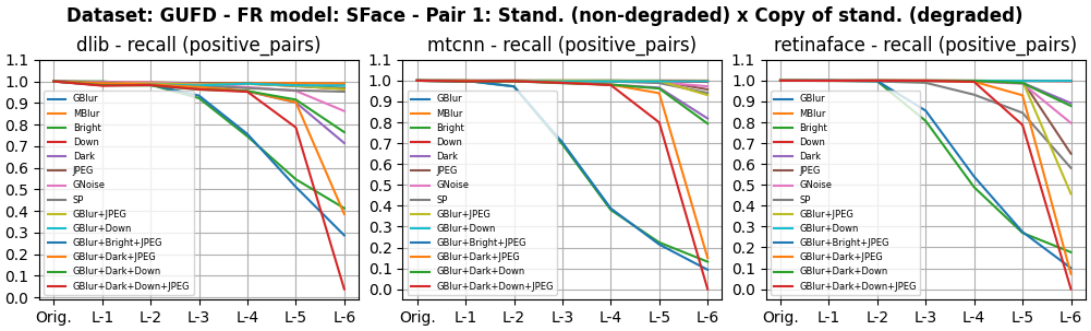


Figura I.55: Dataset GUFD - Par 1 - Métrica *recall* do algoritmo SFace

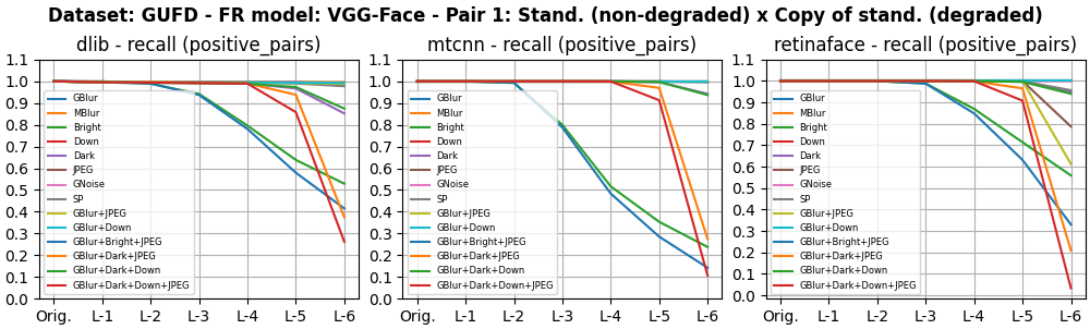


Figura I.56: Dataset GUFD - Par 1 - Métrica *recall* do algoritmo VGG

Dataset GUFD: Par 2 - Imagem padrão (não degradada) x Imagem questionada (degradada)

As imagens deste par são as imagens compreendidas da Figura **??** até a Figura I.64

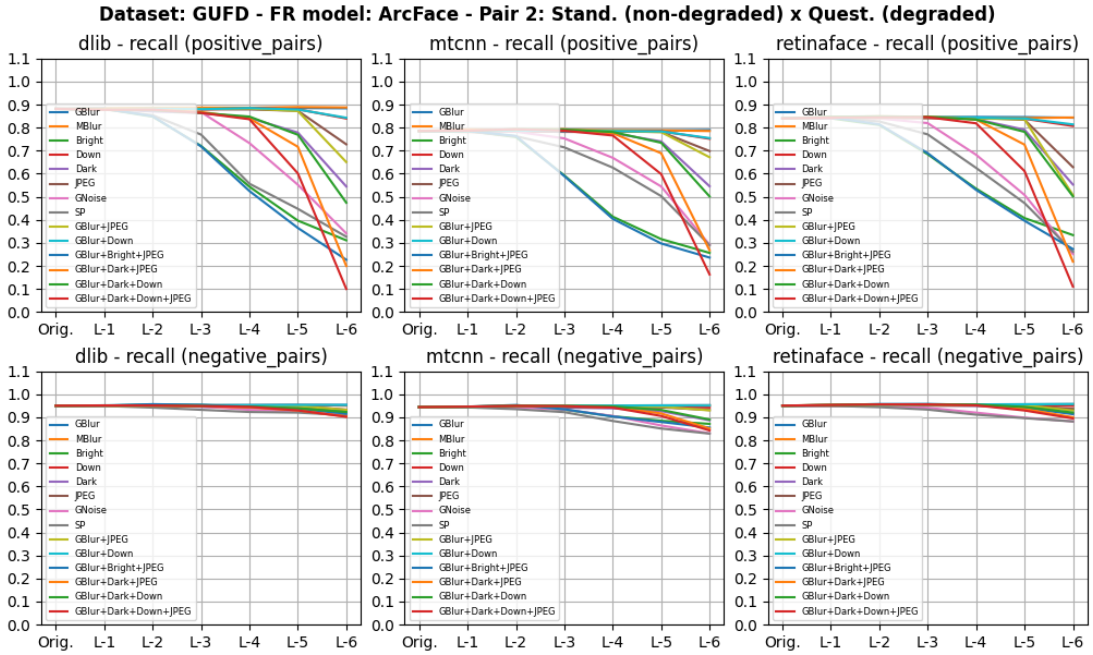


Figura I.57: Dataset GUFD - Par 2 - Métrica *recall* do algoritmo ArcFace

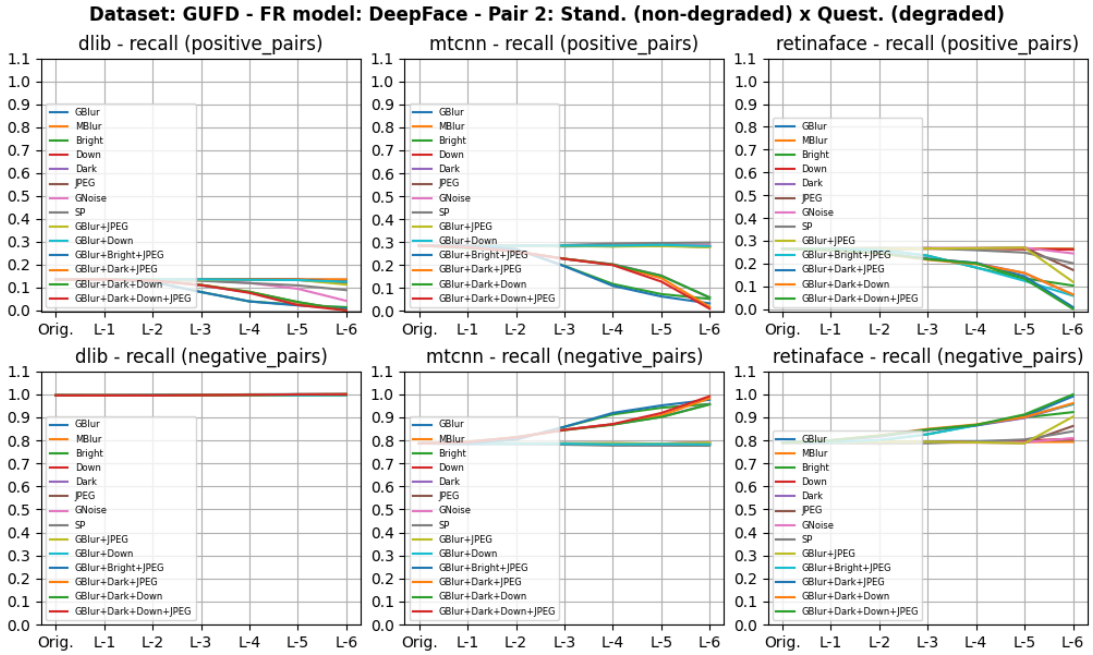


Figura I.58: Dataset GUFD - Par 2 - Métrica *recall* do algoritmo DeepFace

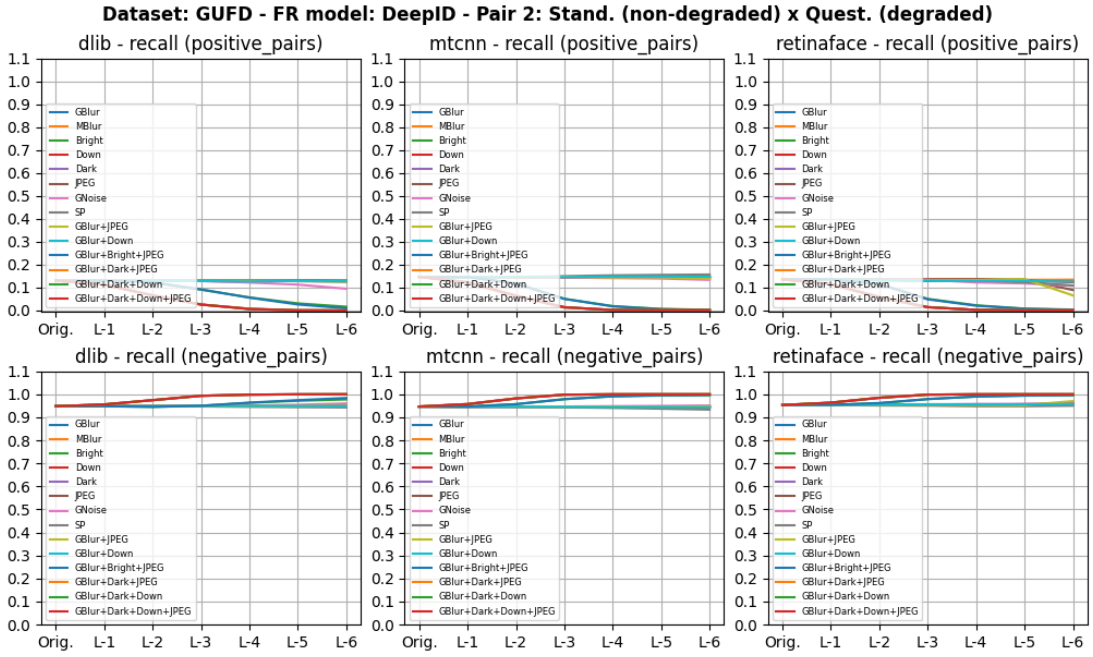


Figura I.59: Dataset GUFD - Par 2 - Métrica *recall* do algoritmo DeepID

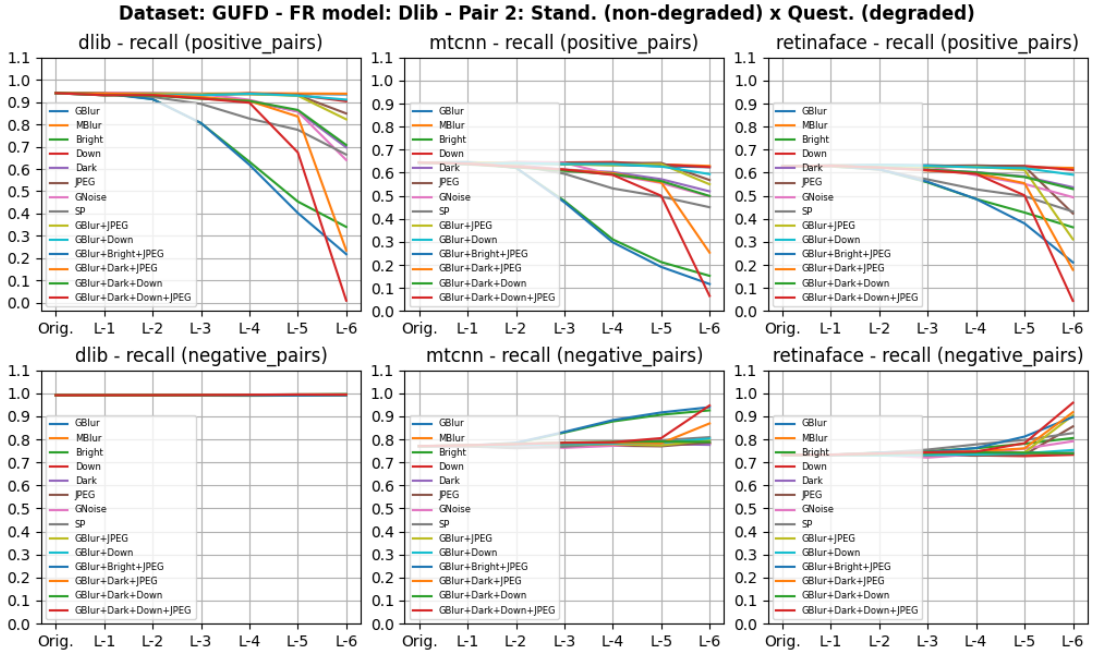


Figura I.60: Dataset GUFD - Par 2 - Métrica *recall* do algoritmo Dlib

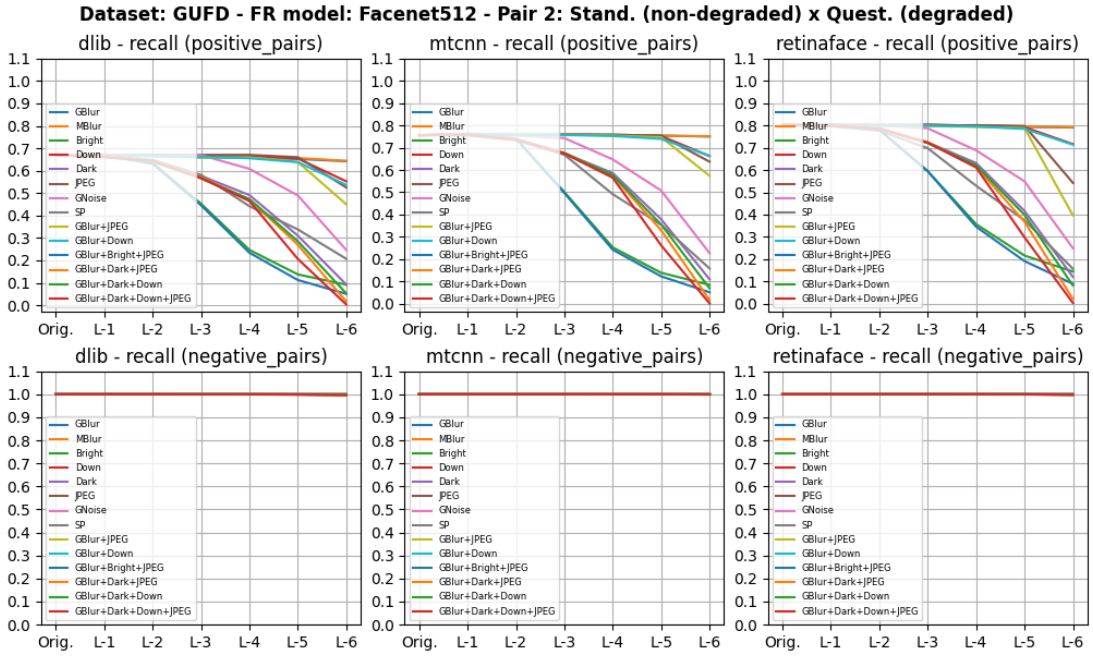


Figura I.61: Dataset GUFD - Par 2 - Métrica *recall* do algoritmo FaceNet512

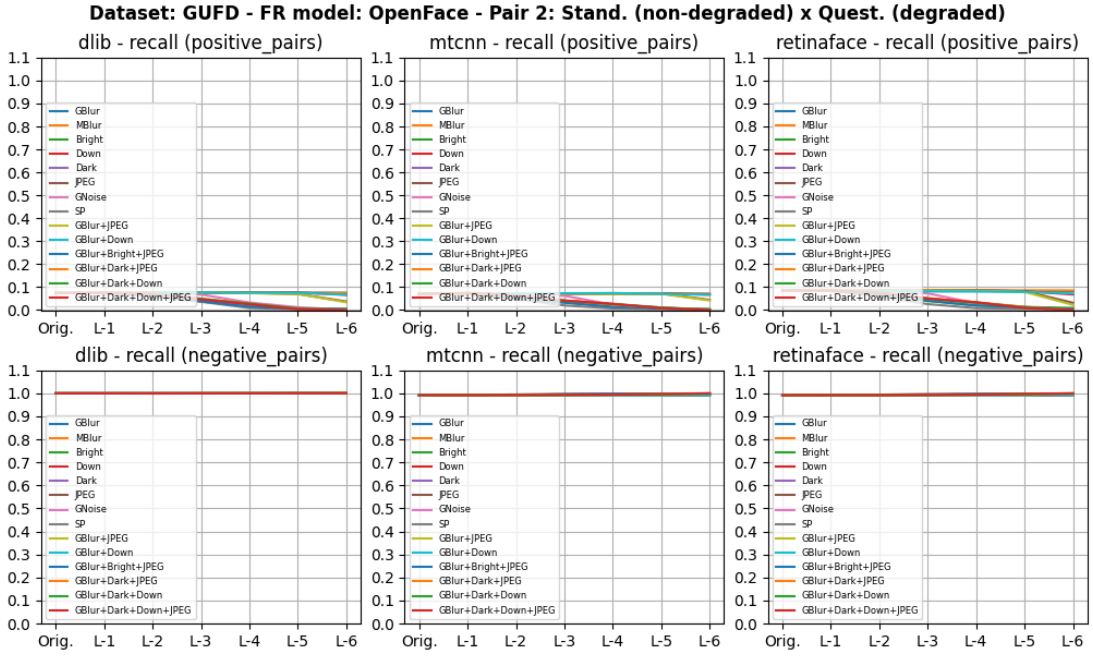


Figura I.62: Dataset GUFD - Par 2 - Métrica *recall* do algoritmo OpenFace

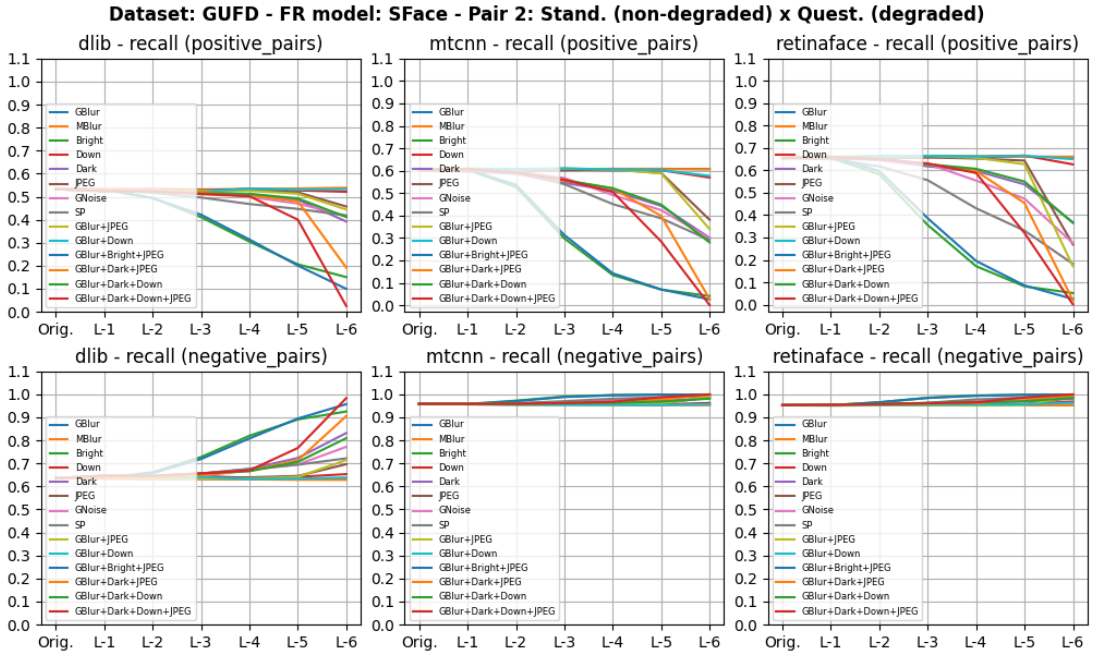


Figura I.63: Dataset GUFD - Par 2 - Métrica *recall* do algoritmo SFace

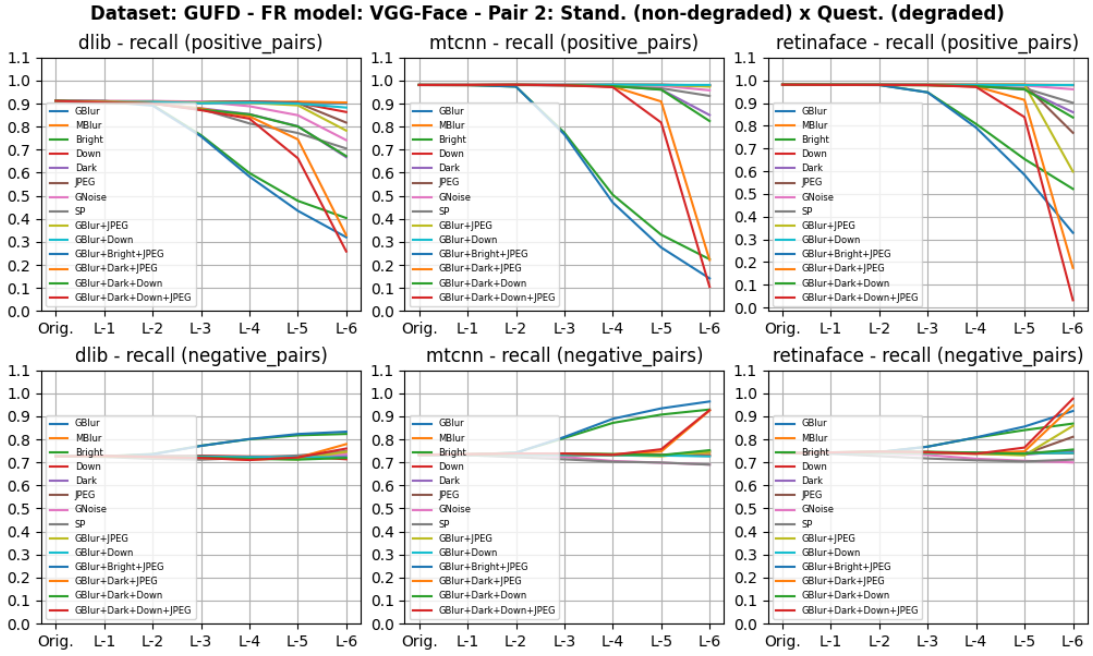


Figura I.64: Dataset GUFD - Par 2 - Métrica *recall* do algoritmo VGG

Dataset GUFD: Par 3 - Imagem padrão (degradada) x Imagem questionada (degradada)

As imagens deste par são as imagens compreendidas da Figura I.65 até a Figura I.72

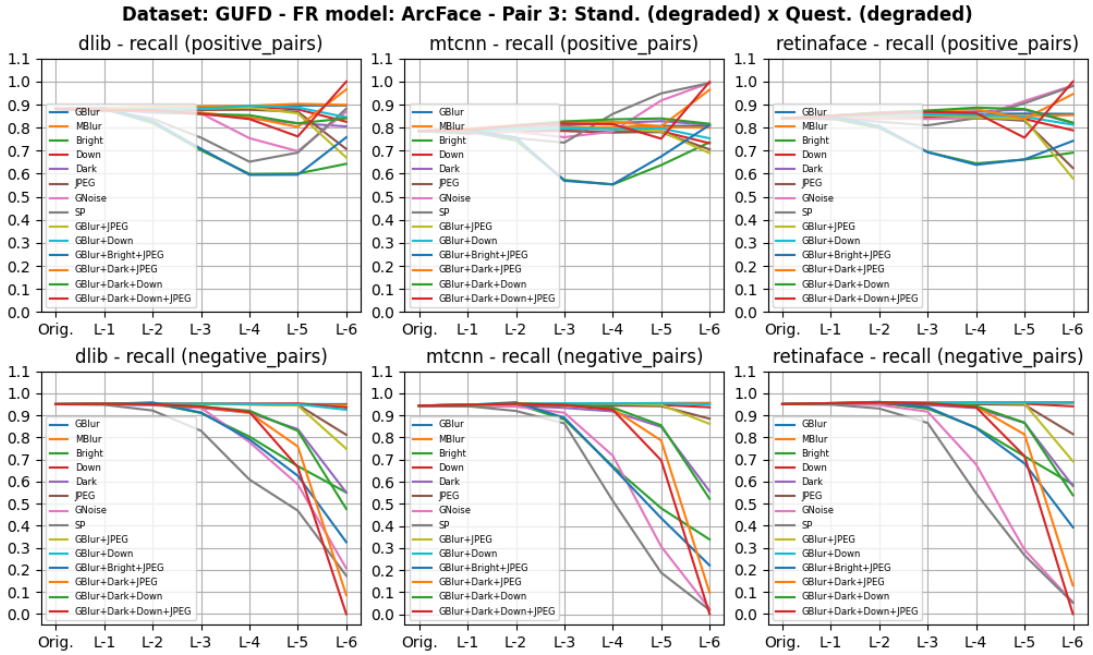


Figura I.65: Dataset GUFD - Par 3 - Métrica *recall* do algoritmo ArcFace

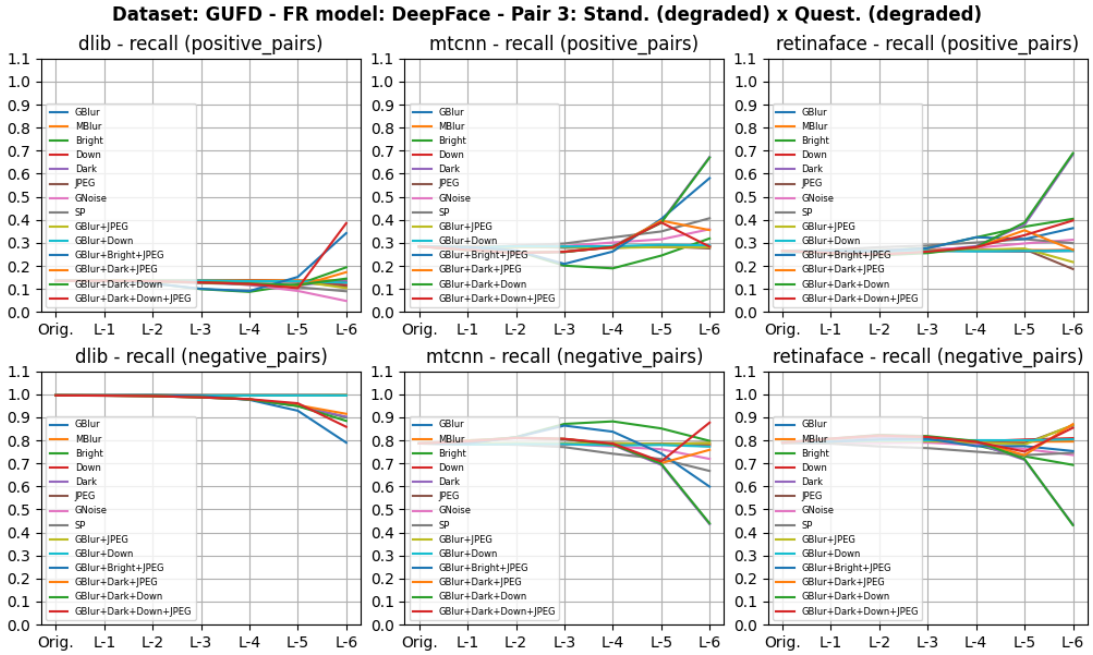


Figura I.66: Dataset GUFD - Par 3 - Métrica *recall* do algoritmo DeepFace

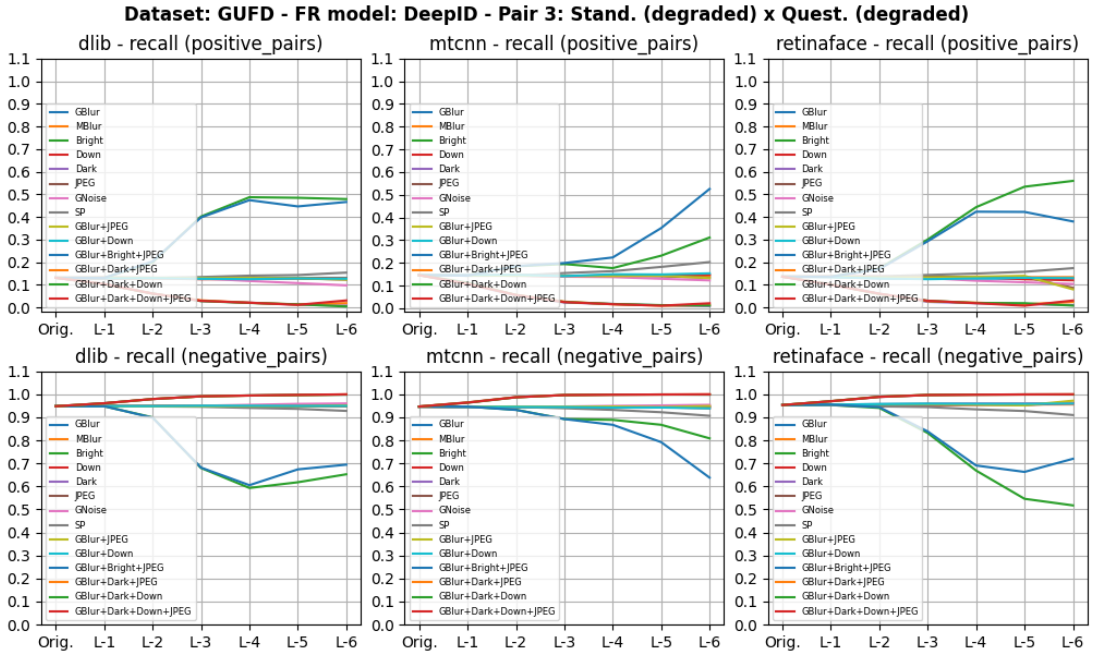


Figura I.67: Dataset GUFD - Par 3 - Métrica *recall* do algoritmo DeepID

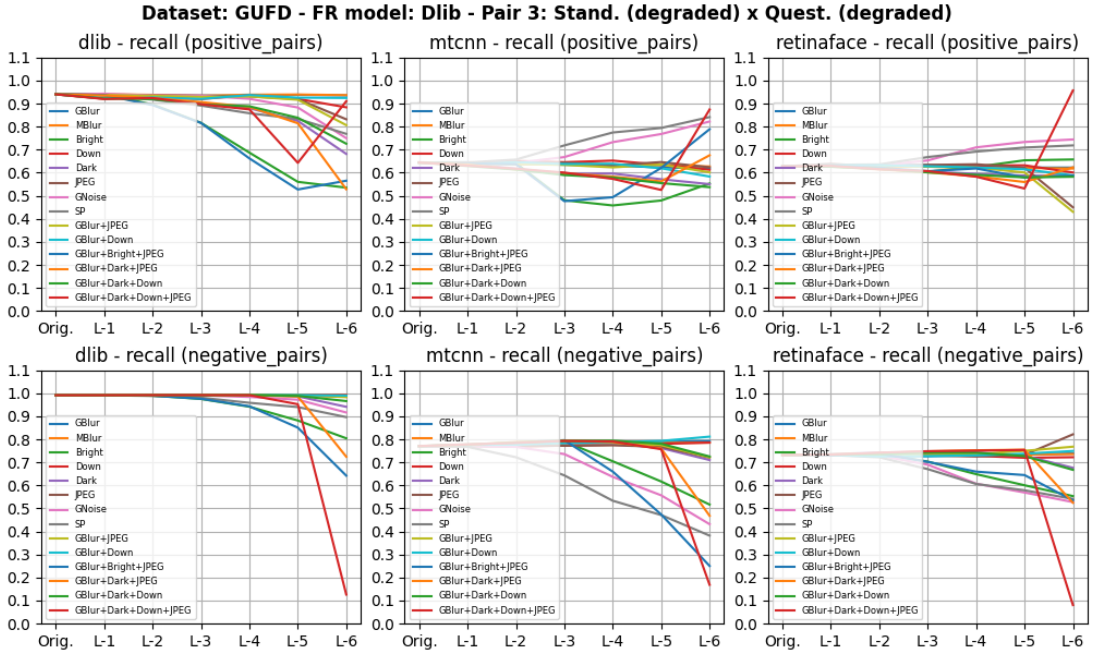


Figura I.68: Dataset GUFD - Par 3 - Métrica *recall* do algoritmo Dlib

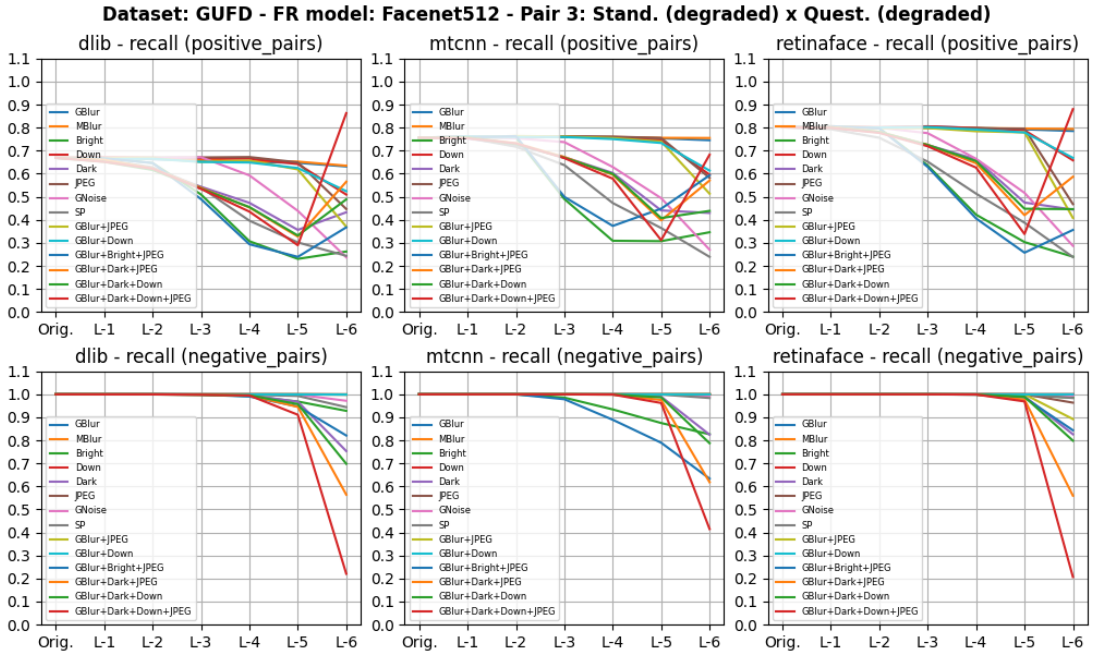


Figura I.69: Dataset GUFD - Par 3 - Métrica *recall* do algoritmo FaceNet512

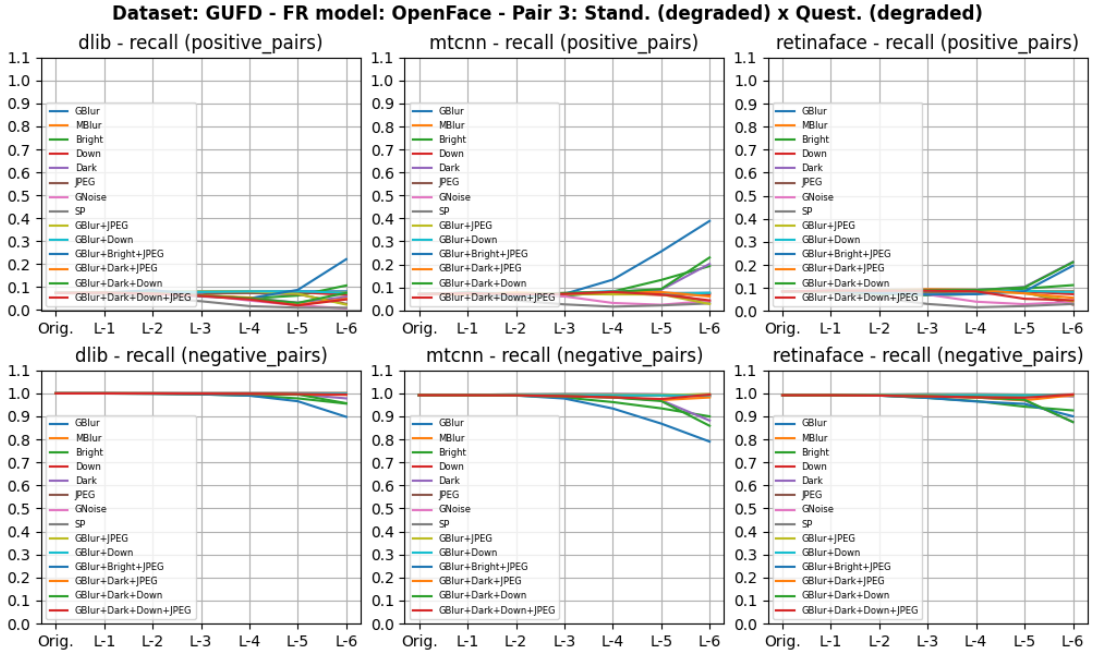


Figura I.70: Dataset GUFD - Par 3 - Métrica *recall* do algoritmo OpenFace

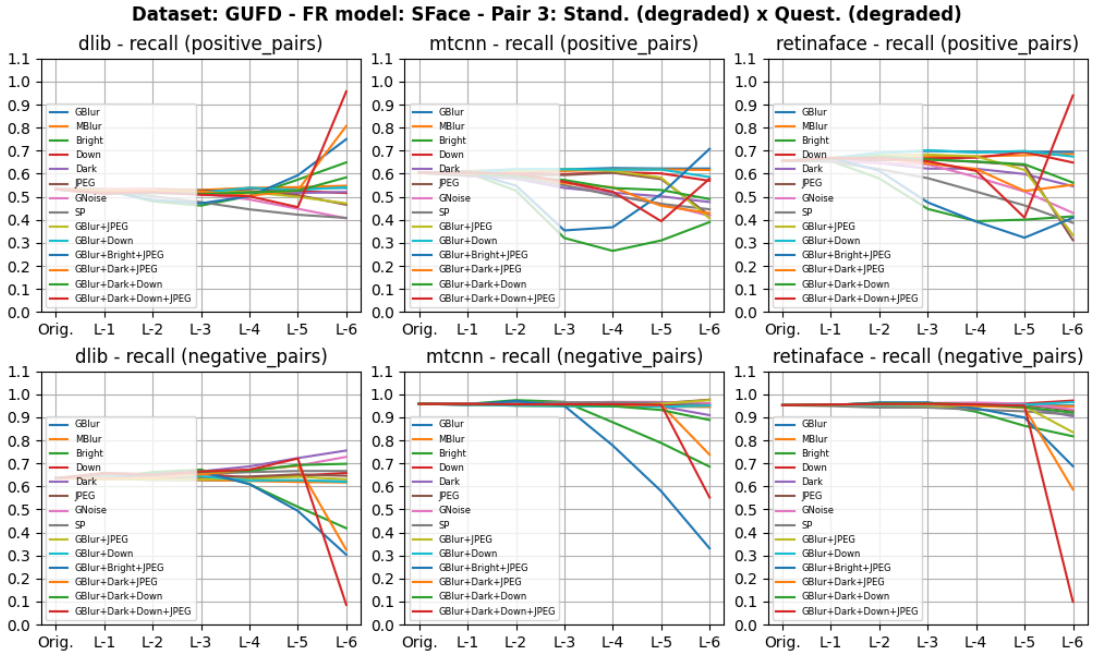


Figura I.71: Dataset GUFD - Par 3 - Métrica *recall* do algoritmo SFace

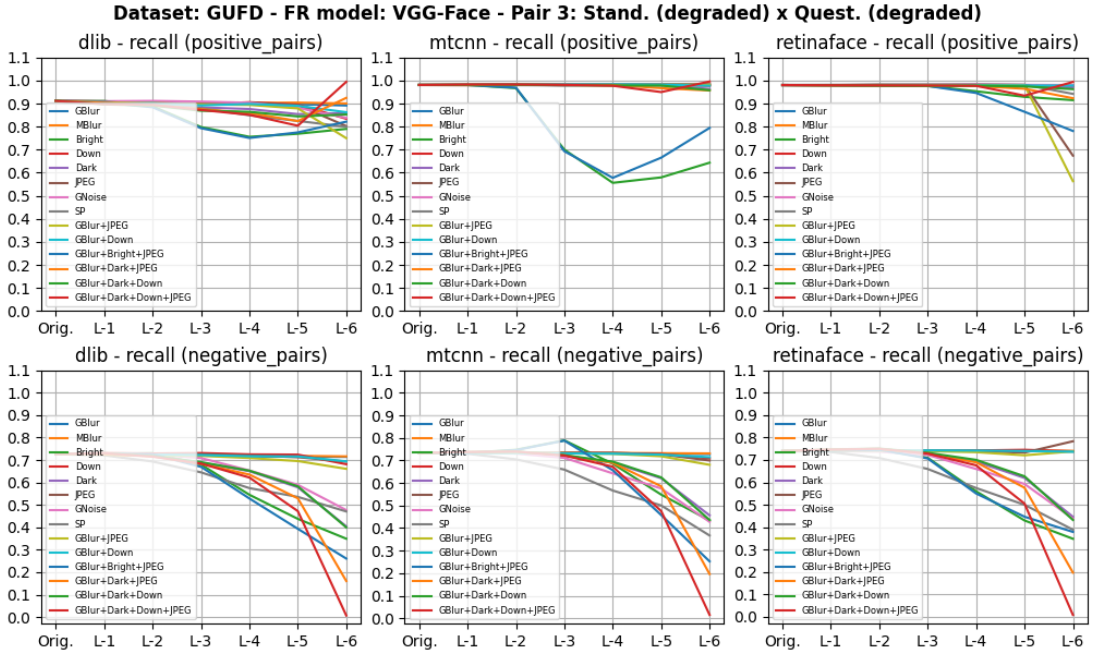


Figura I.72: Dataset GUFD - Par 3 - Métrica *recall* do algoritmo VGG

Dataset SCFace: Par 1 - Imagem padrão (não degradada) x Cópia da imagem padrão (degradada)

As imagens deste par são as imagens compreendidas da Figura I.73 até a Figura I.80

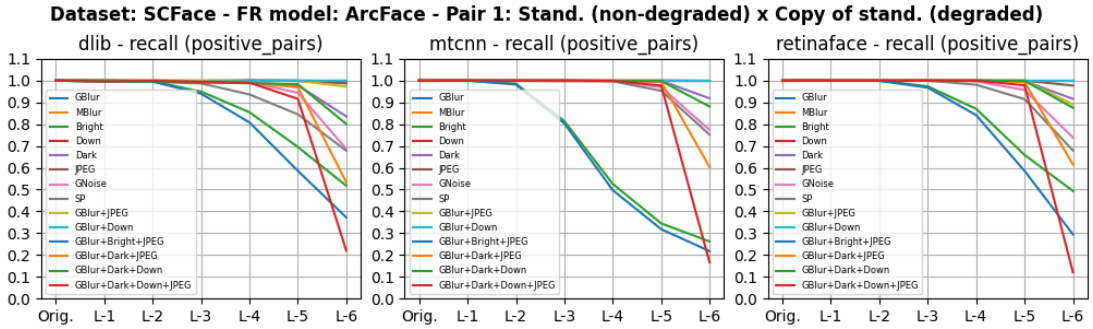


Figura I.73: Dataset SCFace - Par 1 - Métrica *recall* do algoritmo ArcFace

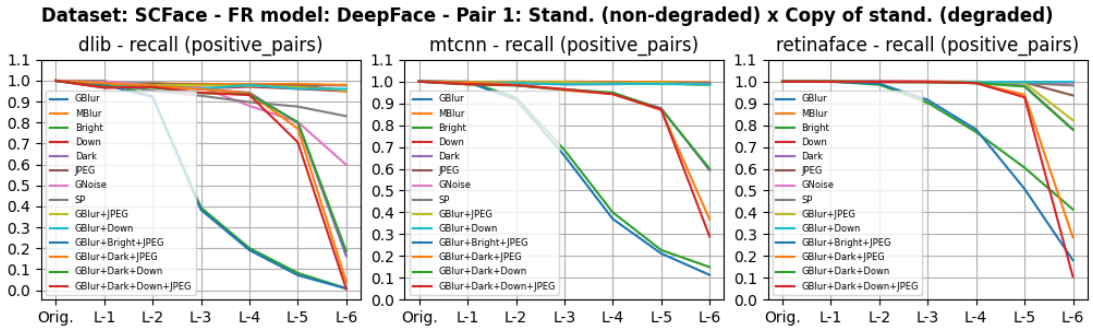


Figura I.74: Dataset SCFace - Par 1 - Métrica *recall* do algoritmo DeepFace

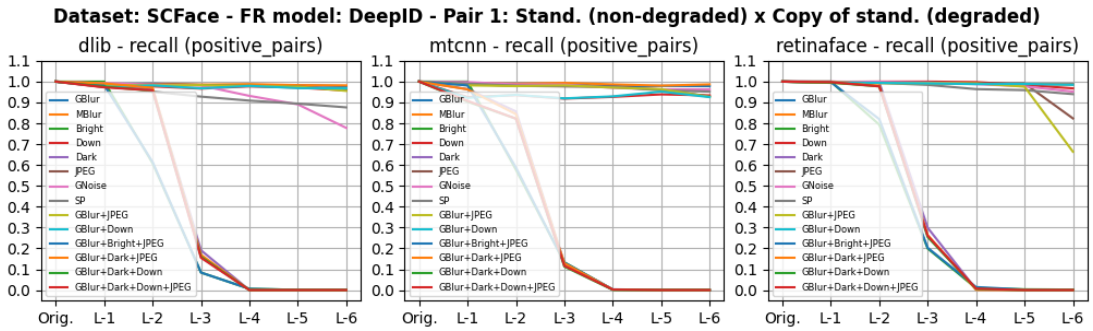


Figura I.75: Dataset SCFace - Par 1 - Métrica *recall* do algoritmo DeepID

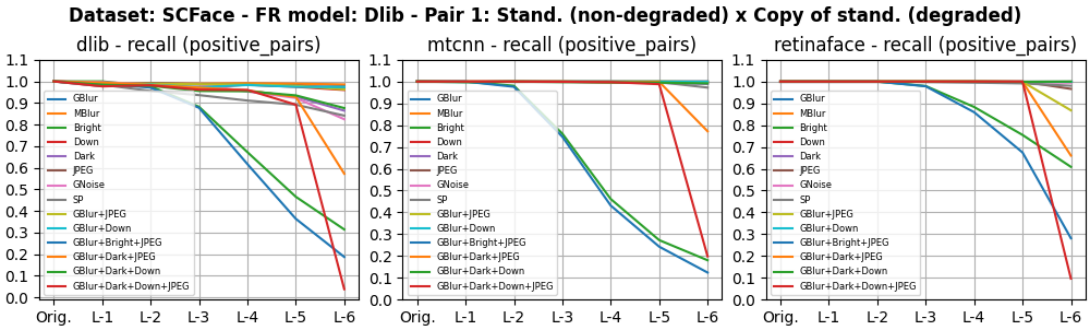


Figura I.76: Dataset SCFace - Par 1 - Métrica *recall* do algoritmo Dlib

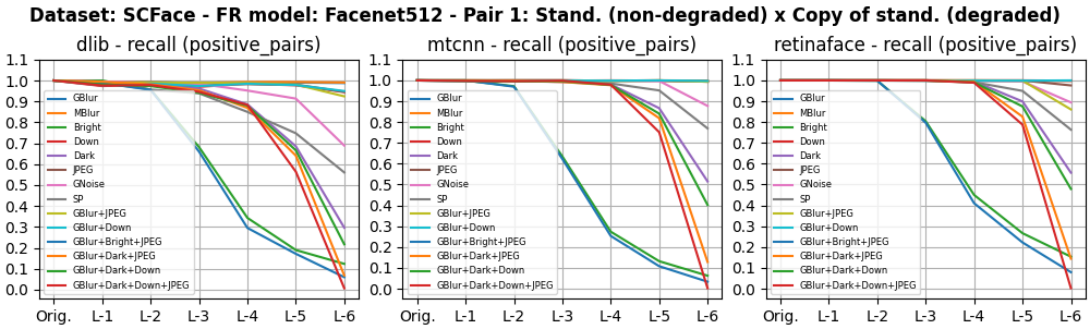


Figura I.77: Dataset SCFace - Par 1 - Métrica *recall* do algoritmo FaceNet512

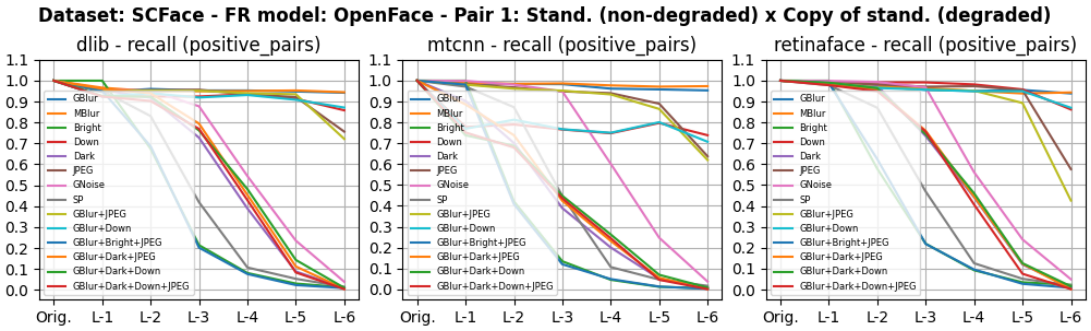


Figura I.78: Dataset SCFace - Par 1 - Métrica *recall* do algoritmo OpenFace

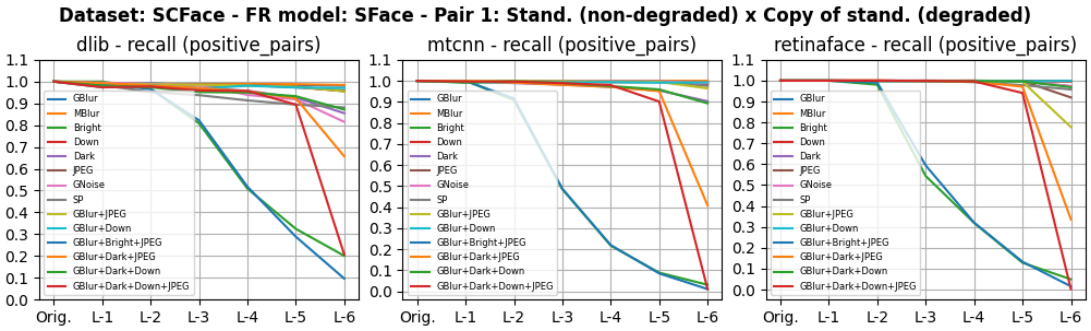


Figura I.79: Métrica *recall* do algoritmo SFace

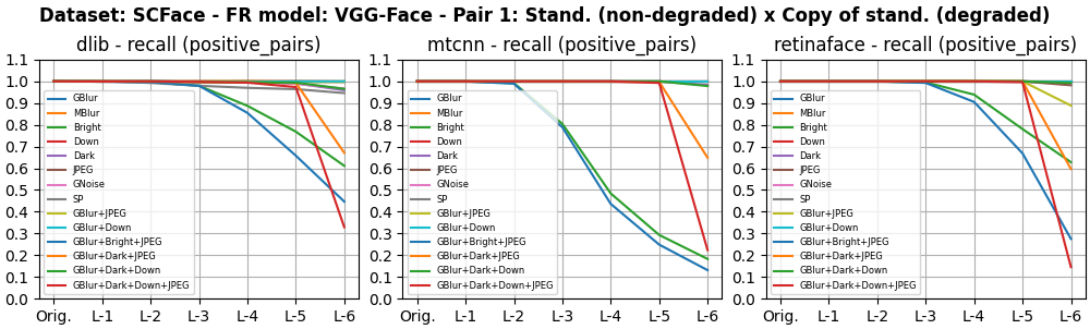


Figura I.80: Dataset SCFace - Par 1 - Métrica *recall* do algoritmo VGG

Dataset SCFace: Par 2 - Imagem padrão (não degradada) x Imagem questionada (degradada)

As imagens deste par são as imagens compreendidas da Figura I.81 até a Figura I.88

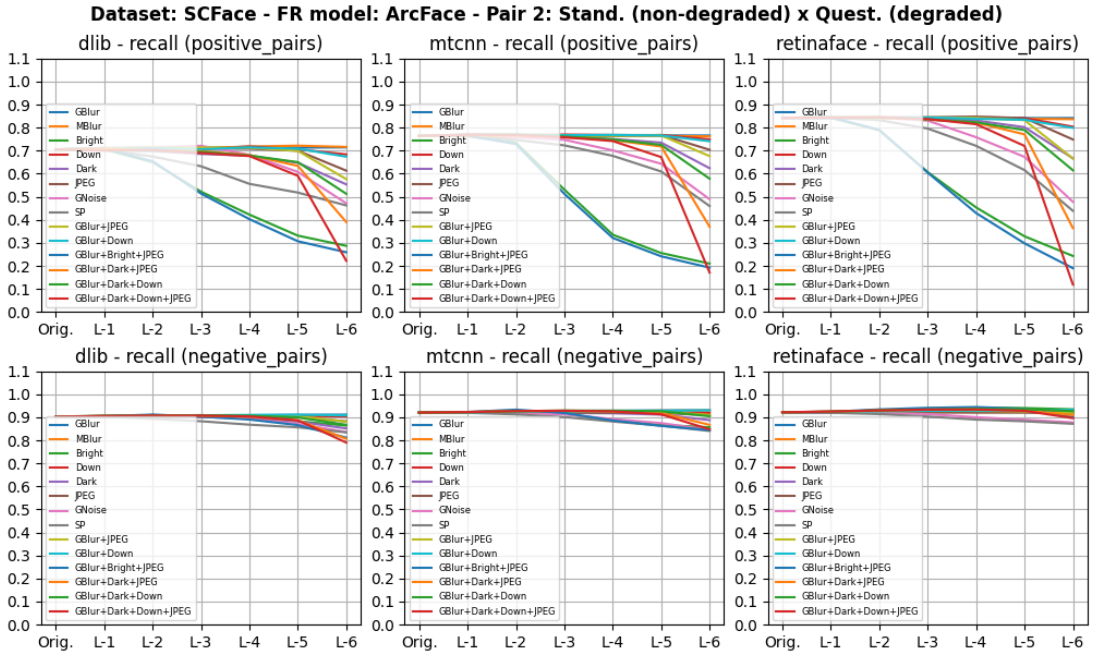


Figura I.81: Dataset SCFace - Par 2 - Métrica *recall* do algoritmo ArcFace

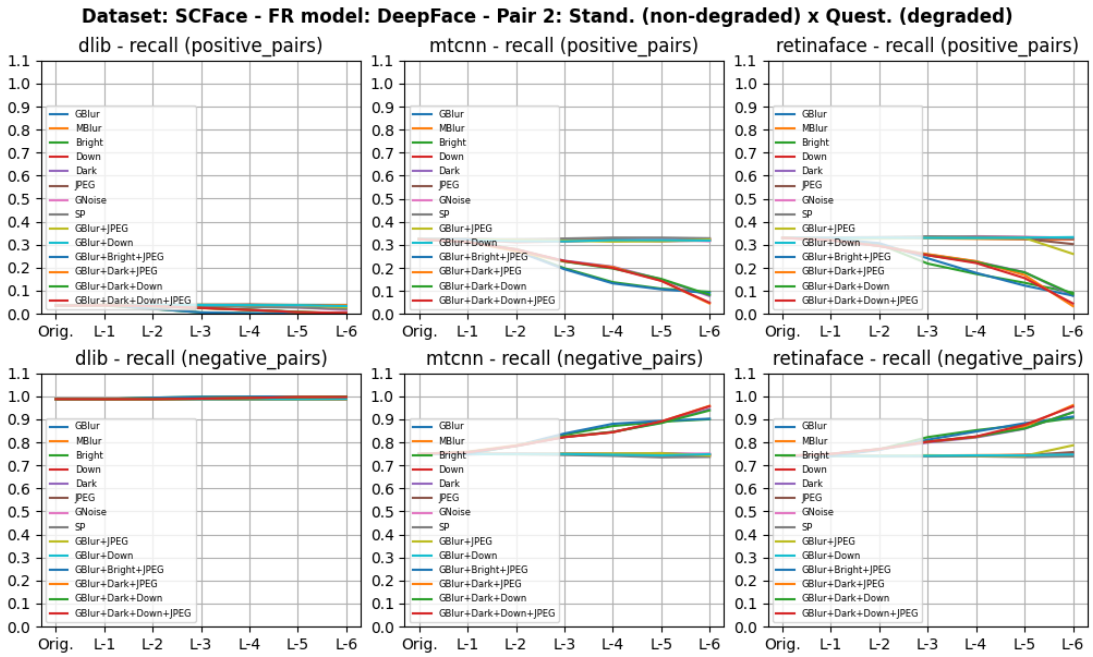


Figura I.82: Dataset SCFace - Par 2 - Métrica *recall* do algoritmo DeepFace

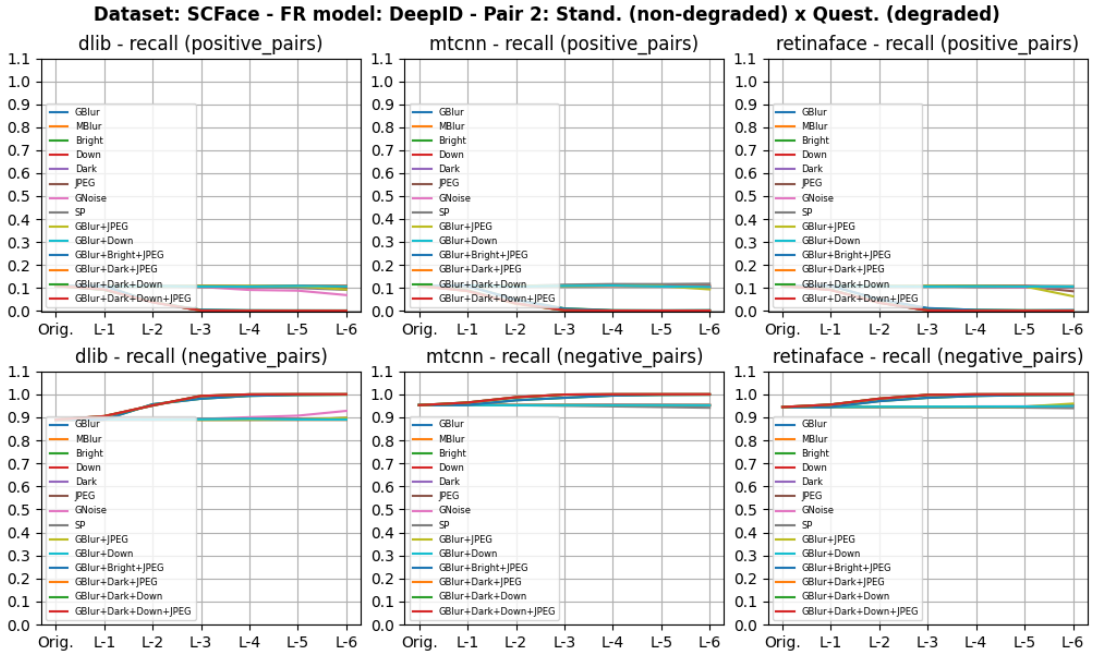


Figura I.83: Dataset SCFace - Par 2 - Métrica *recall* do algoritmo DeepID

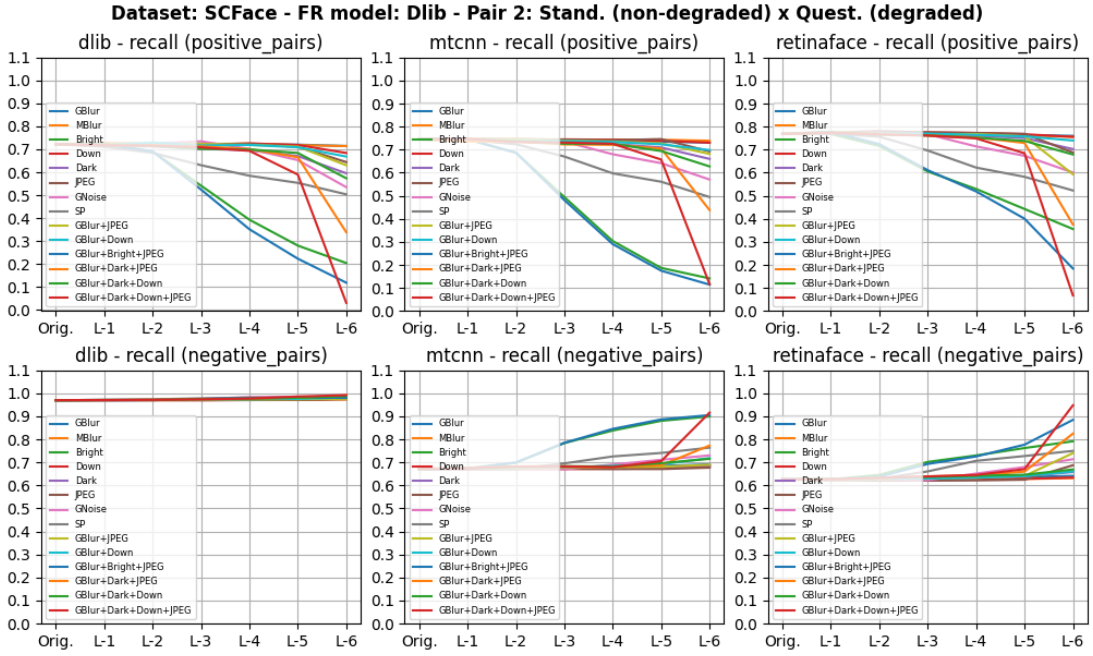


Figura I.84: Dataset SCFace - Par 2 - Métrica *recall* do algoritmo Dlib

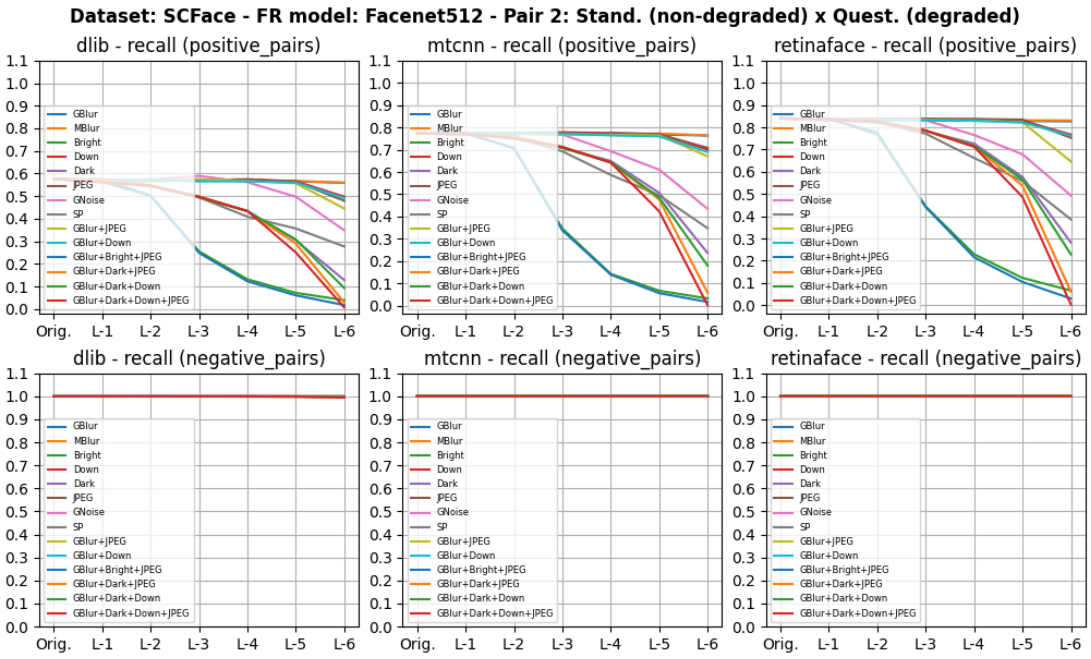


Figura I.85: Dataset SCFace - Par 2 - Métrica *recall* do algoritmo FaceNet512

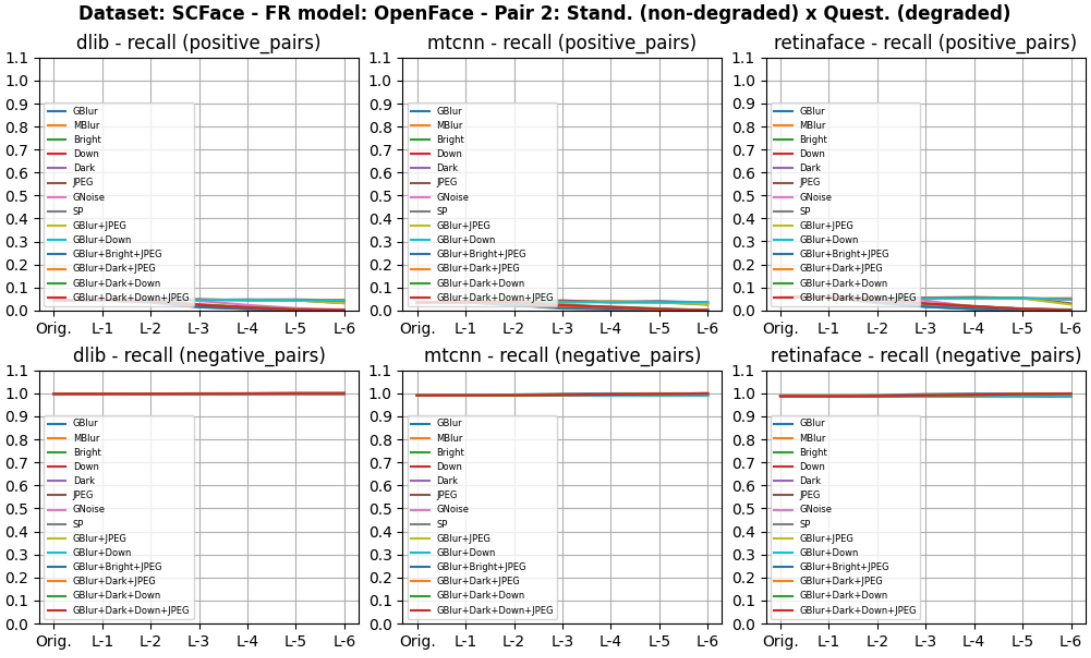


Figura I.86: Dataset SCFace - Par 2 - Métrica *recall* do algoritmo OpenFace

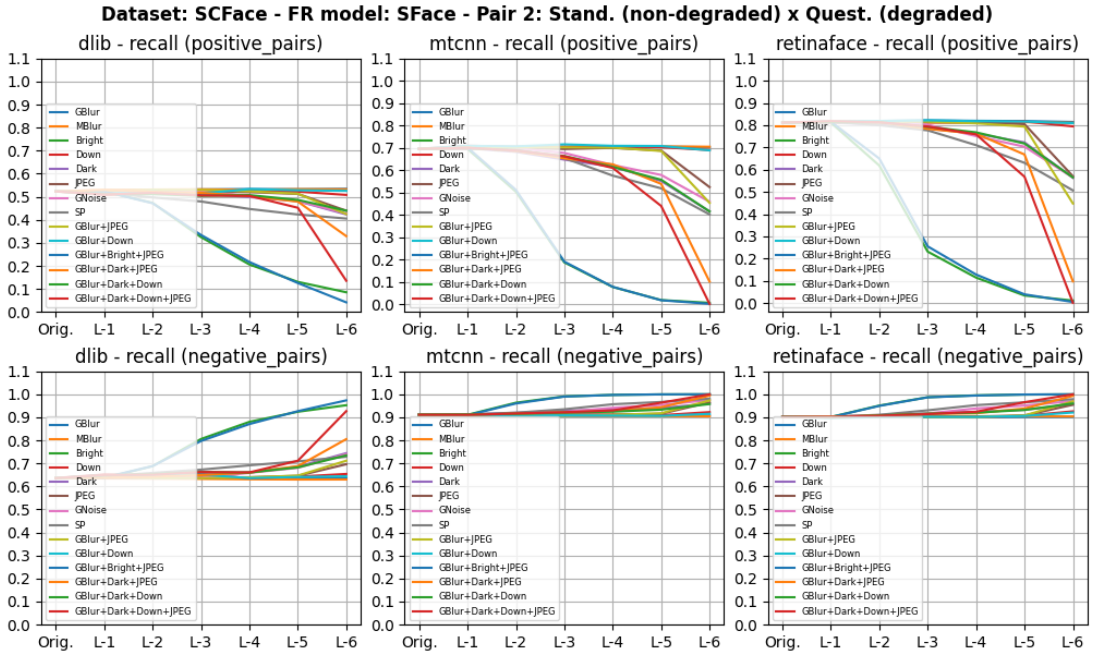


Figura I.87: Dataset SCFace - Par 2 - Métrica *recall* do algoritmo SFace

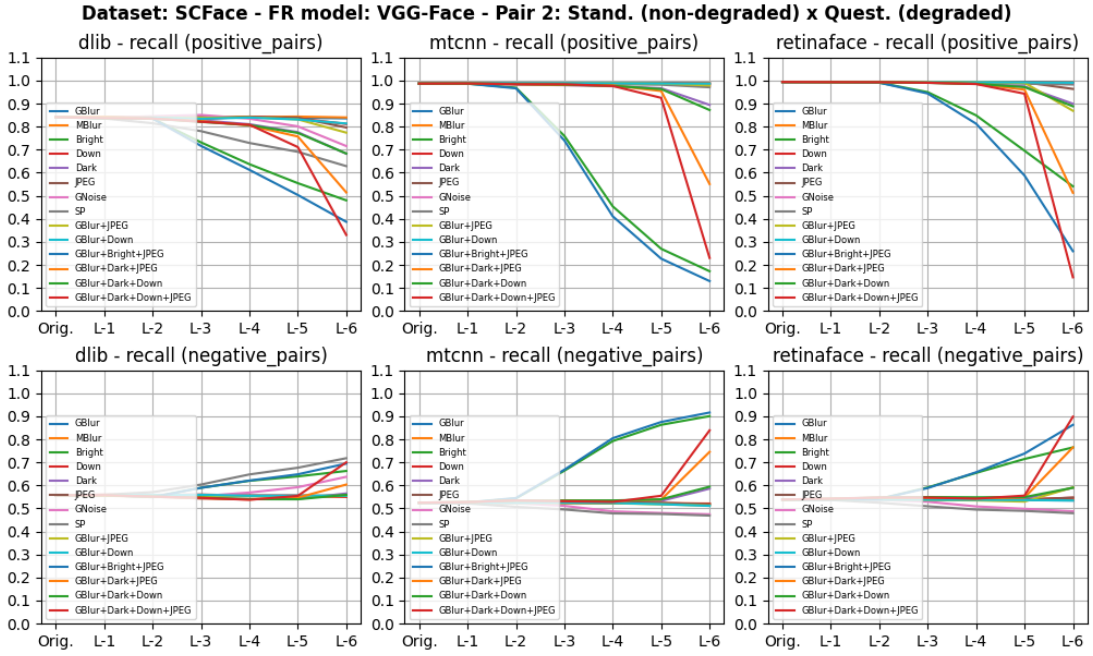


Figura I.88: Dataset SCFace - Par 2 - Métrica *recall* do algoritmo VGG

Dataset SCFace: Par 3 - Imagem padrão (degradada) x Imagem questionada (degradada)

As imagens deste par são as imagens compreendidas da Figura I.89 até a Figura I.96

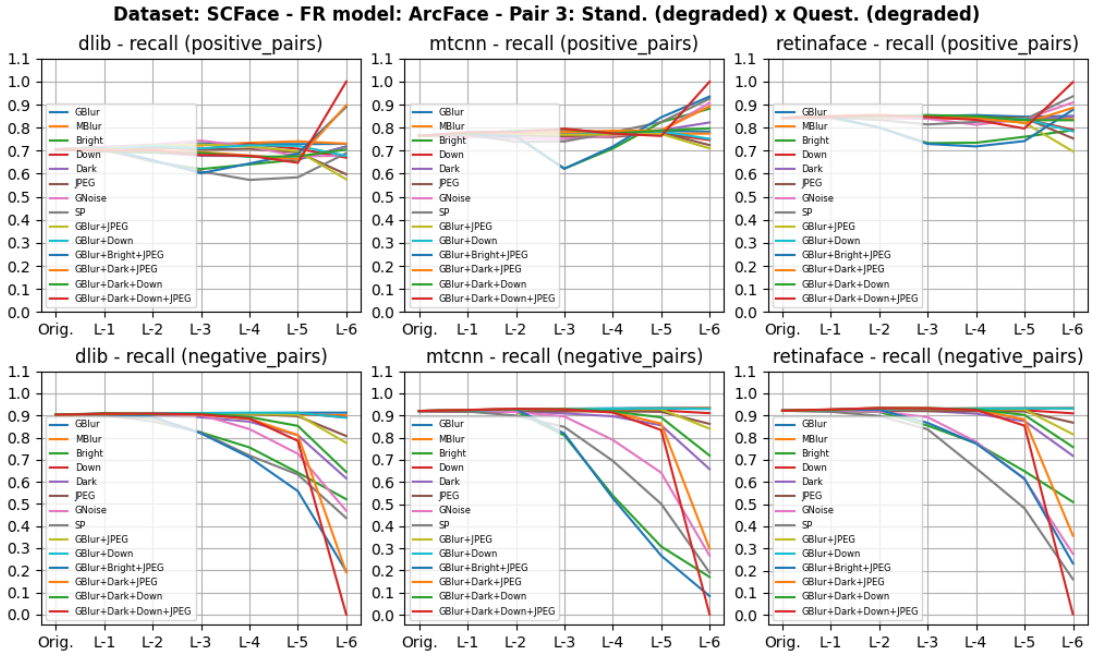


Figura I.89: Dataset SCFace - Par 3 - Métrica *recall* do algoritmo ArcFace

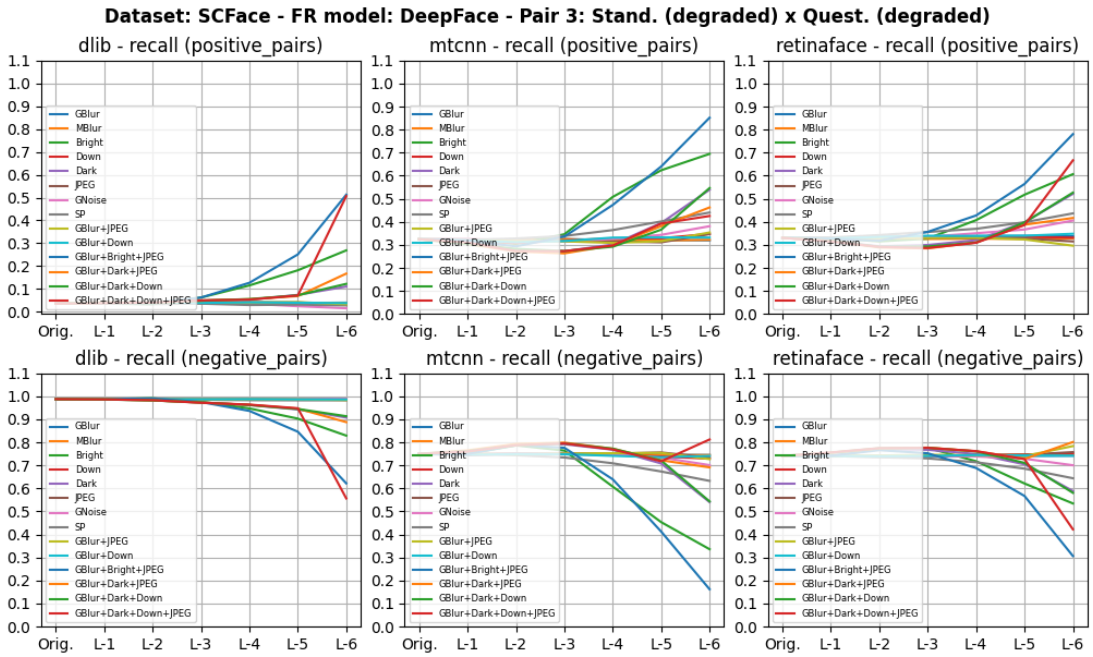


Figura I.90: Dataset SCFace - Par 3 - Métrica *recall* do algoritmo DeepFace

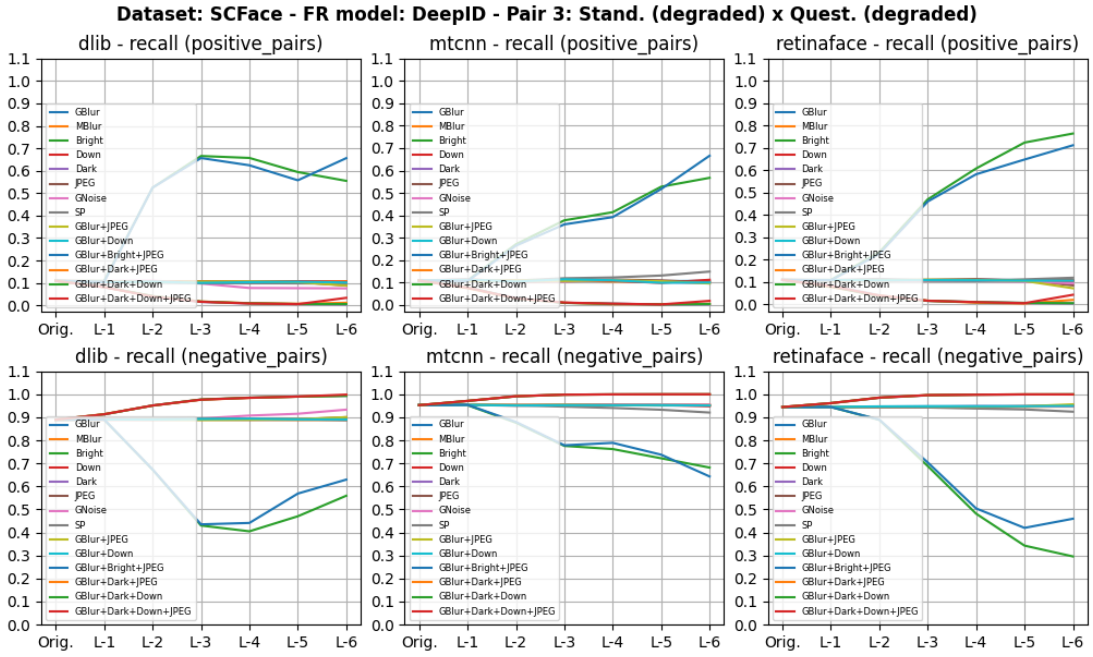


Figura I.91: Dataset SCFace - Par 3 - Métrica *recall* do algoritmo DeepID

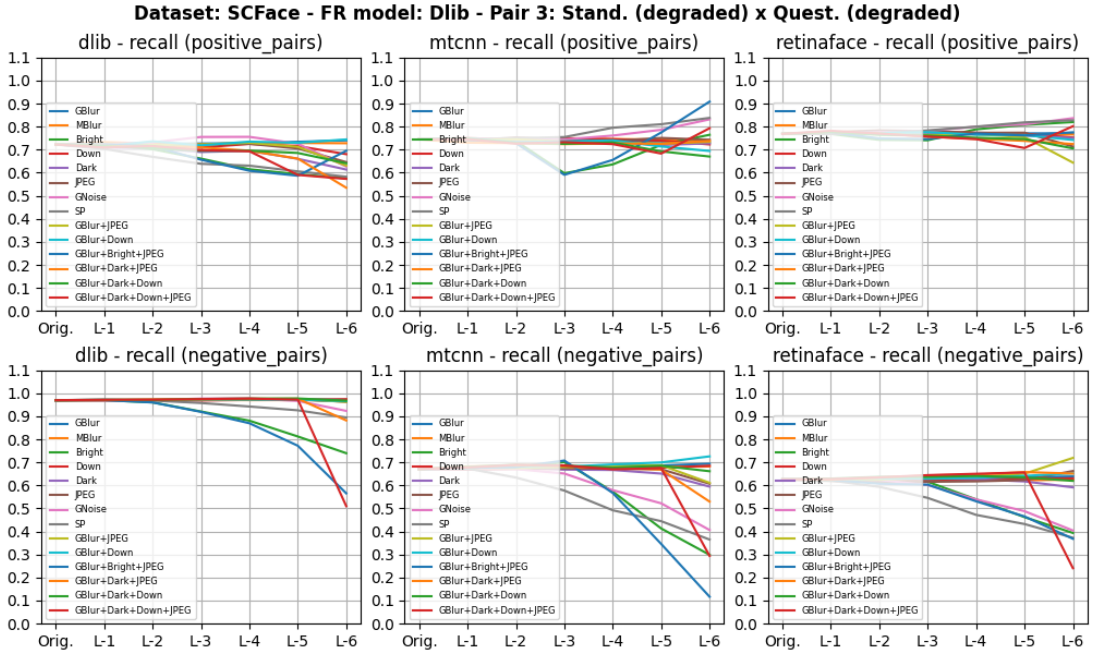


Figura I.92: Dataset SCFace - Par 3 - Métrica *recall* do algoritmo Dlib

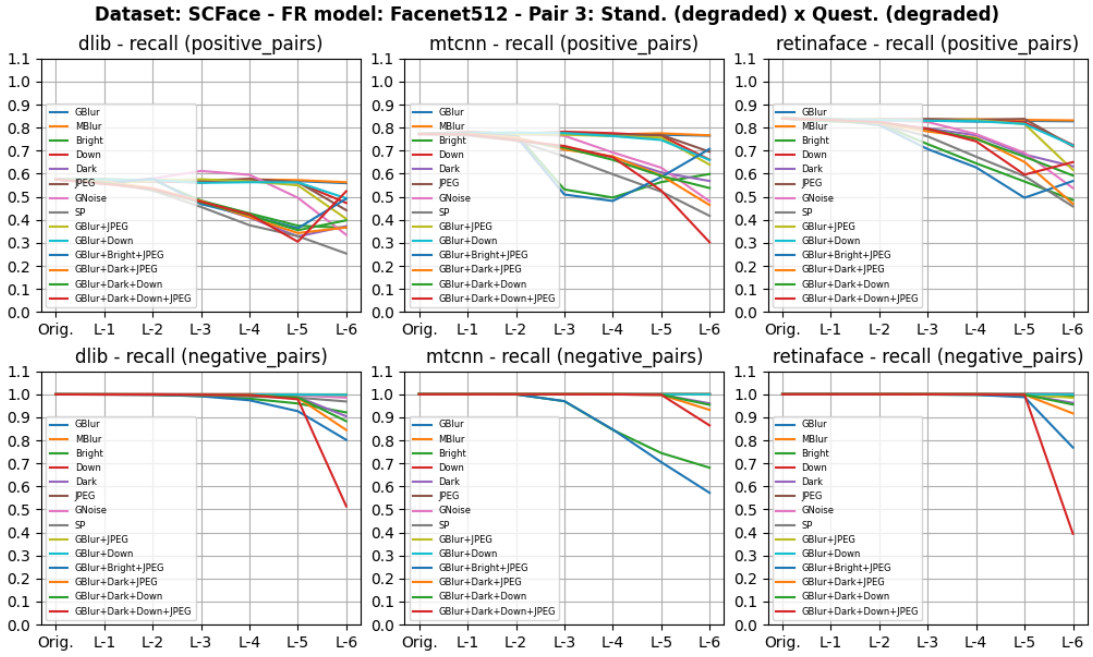


Figura I.93: Dataset SCFace - Par 3 - Métrica *recall* do algoritmo FaceNet512

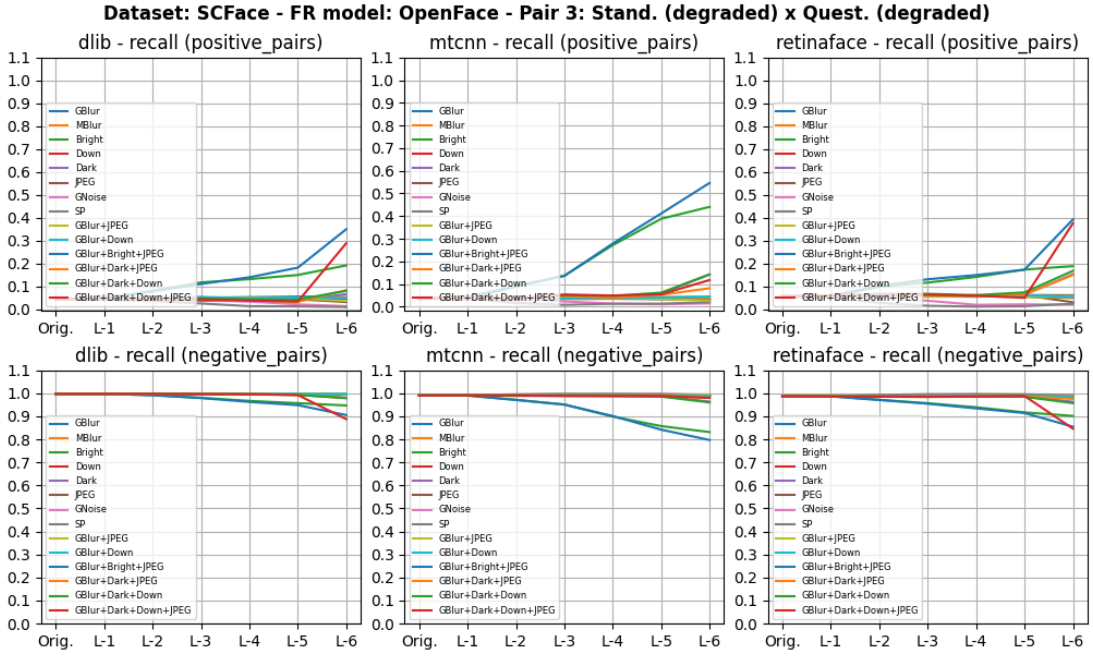


Figura I.94: Dataset SCFace - Par 3 - Métrica *recall* do algoritmo OpenFace

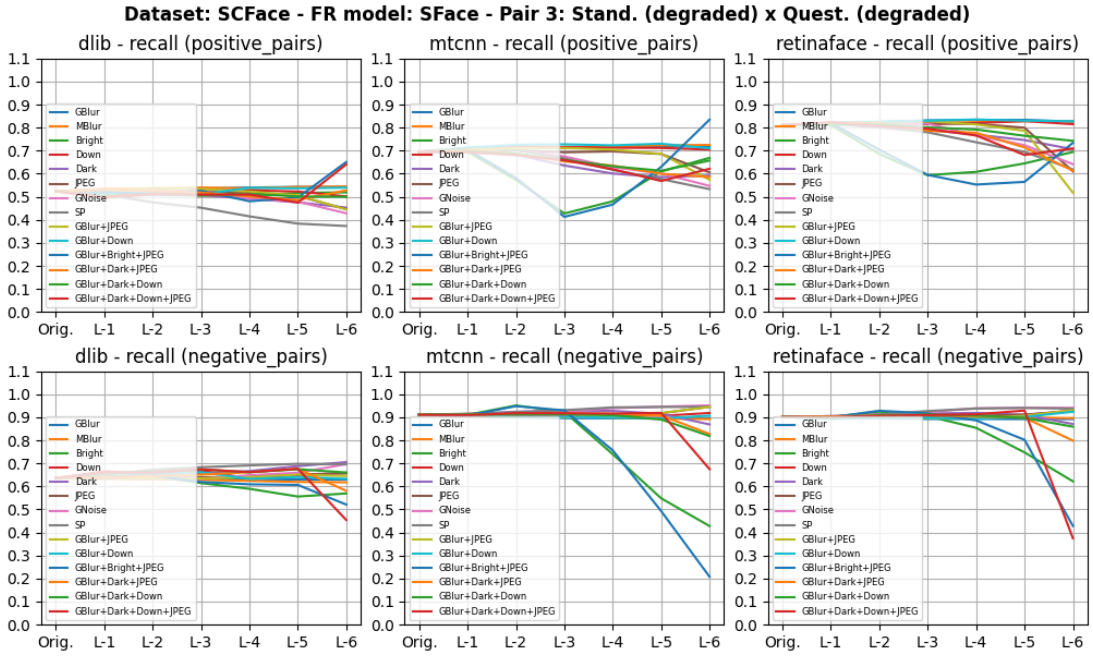


Figura I.95: Dataset SCFace - Par 3 - Métrica *recall* do algoritmo SFace

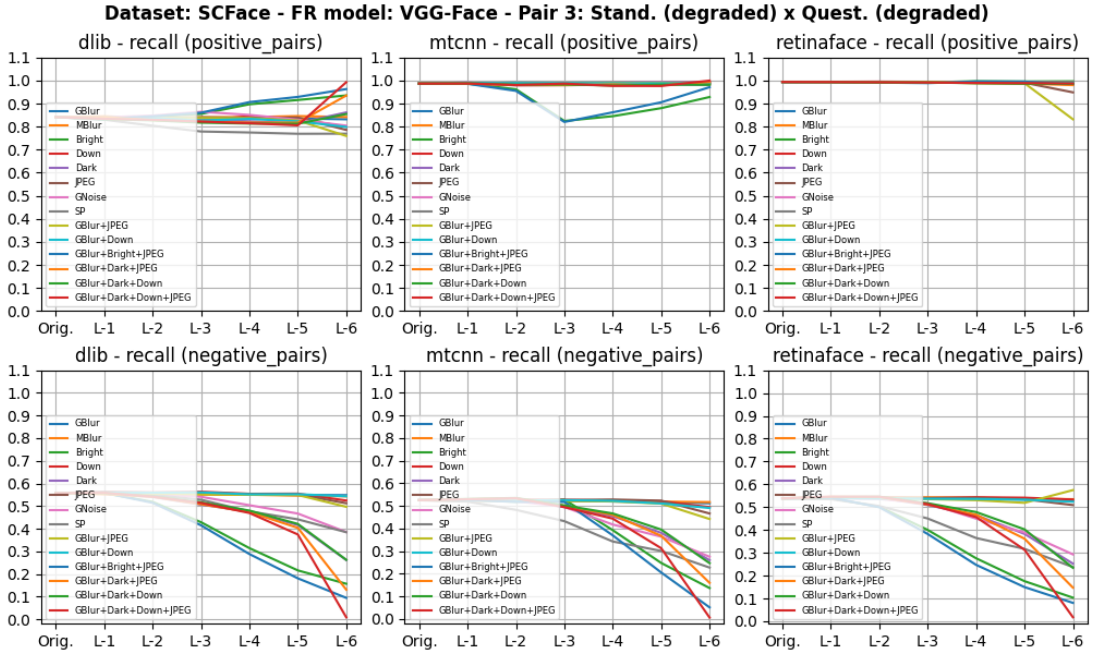


Figura I.96: Dataset SCFace - Par 3 - Métrica *recall* do algoritmo VGG

I.2 Algoritmos de detecção facial

Os resultados foram subdivididos por *dataset*. Ainda, cada gráfico exibe o desempenho dos 3 algoritmos de detecção facial estudados, para determinado tipo de degradação.

Dataset LFW

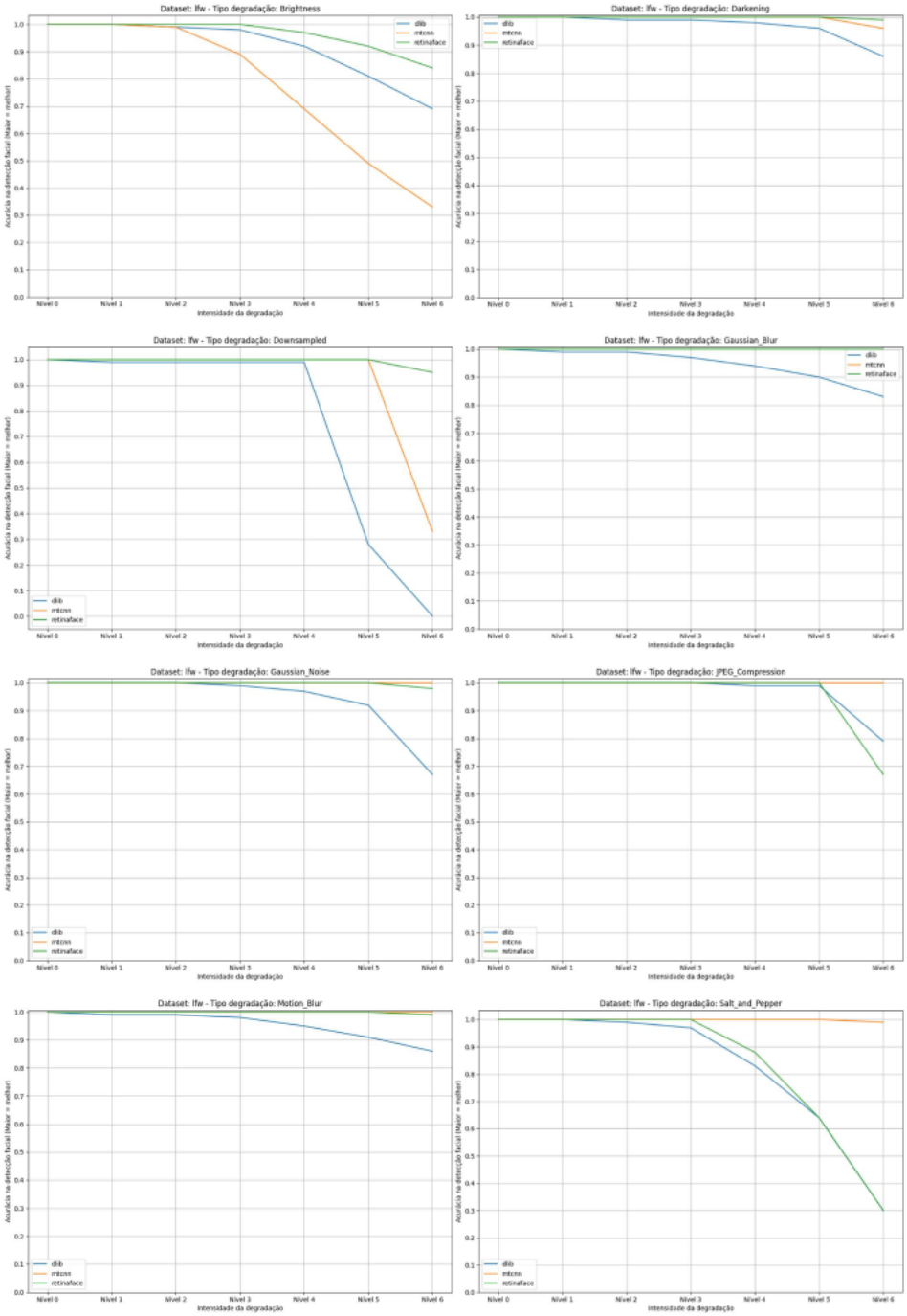


Figura I.97: Desempenho dos algoritmos de detecção facial (degradações simples) no *dataset* LFW

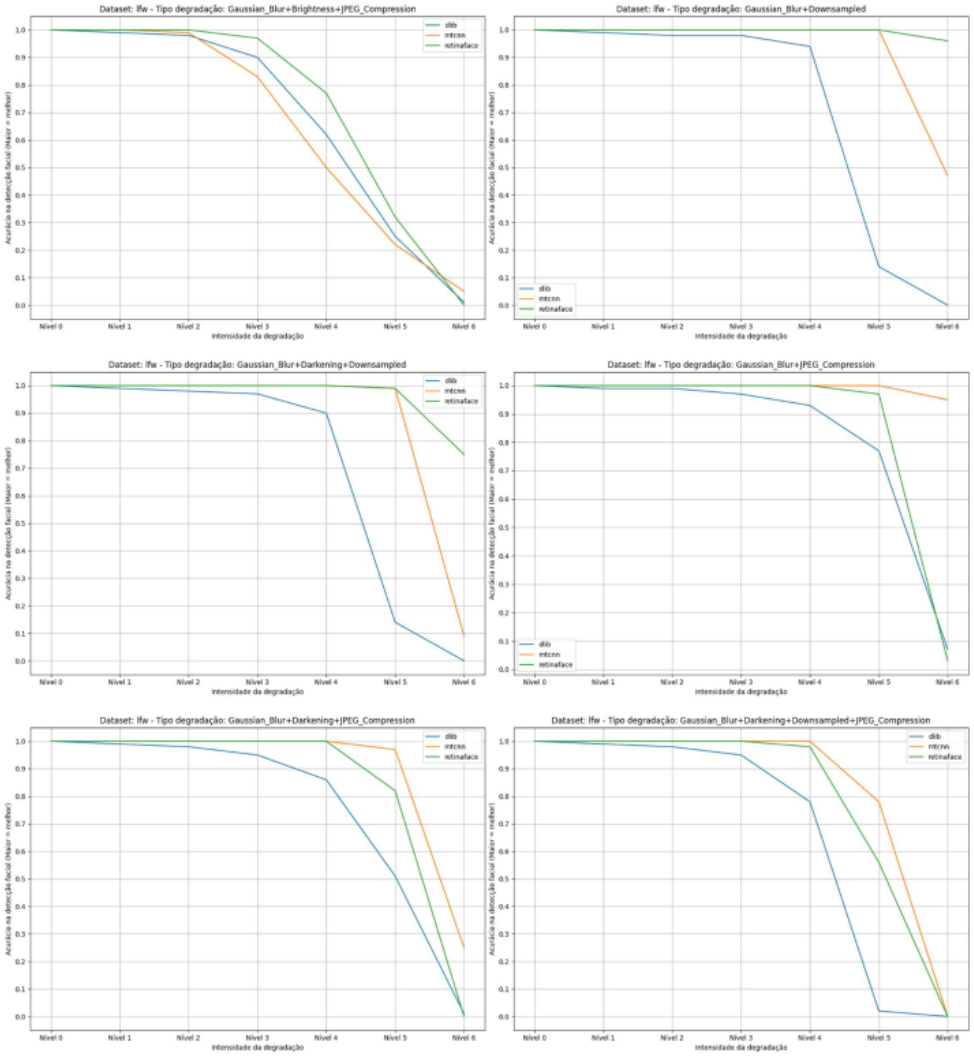


Figura I.98: Desempenho dos algoritmos de detecção facial (degradações em sequência) no *dataset* LFW

Dataset FEI

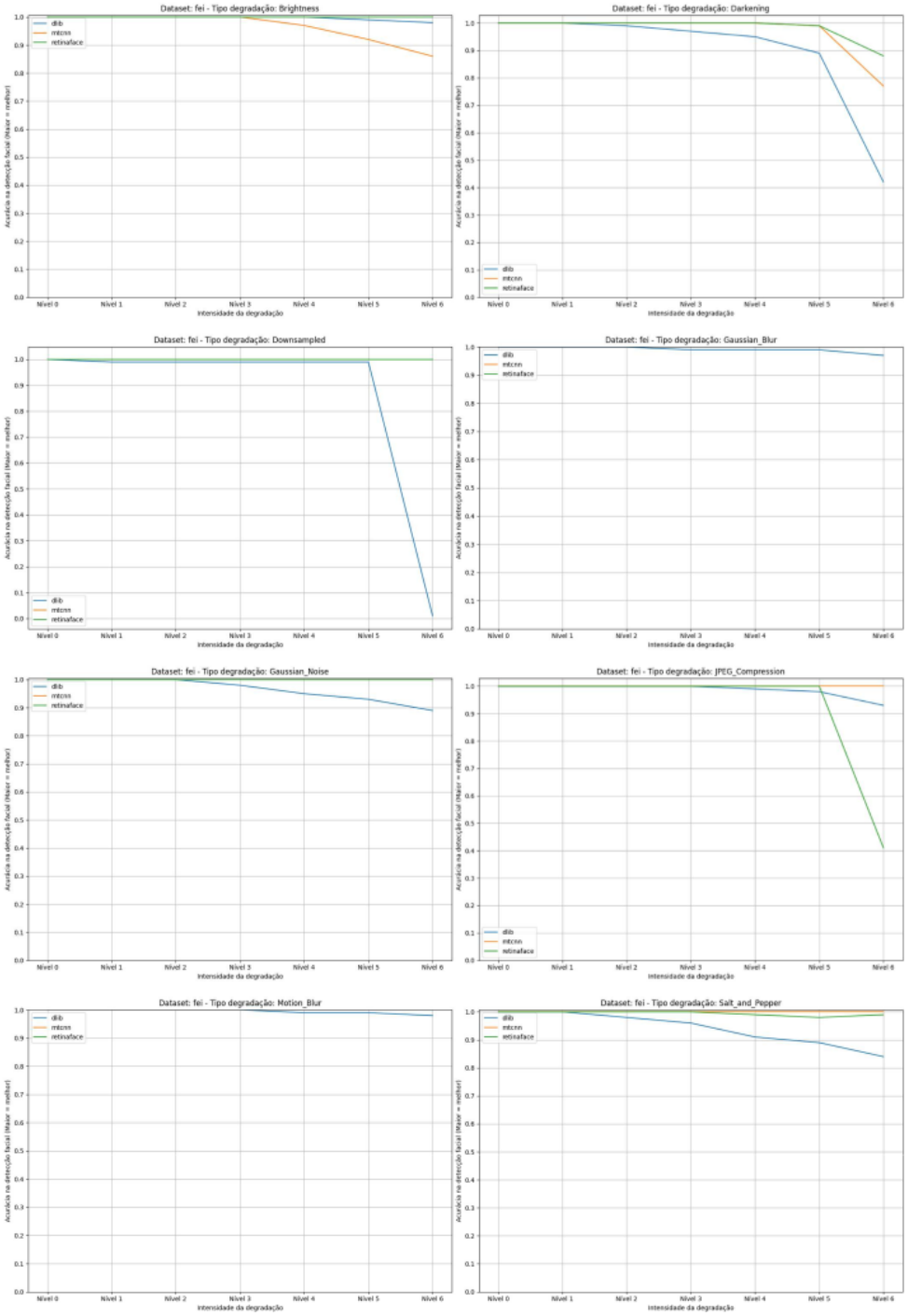


Figura I.99: Desempenho dos algoritmos de detecção facial (degradações simples) no *dataset* FEI

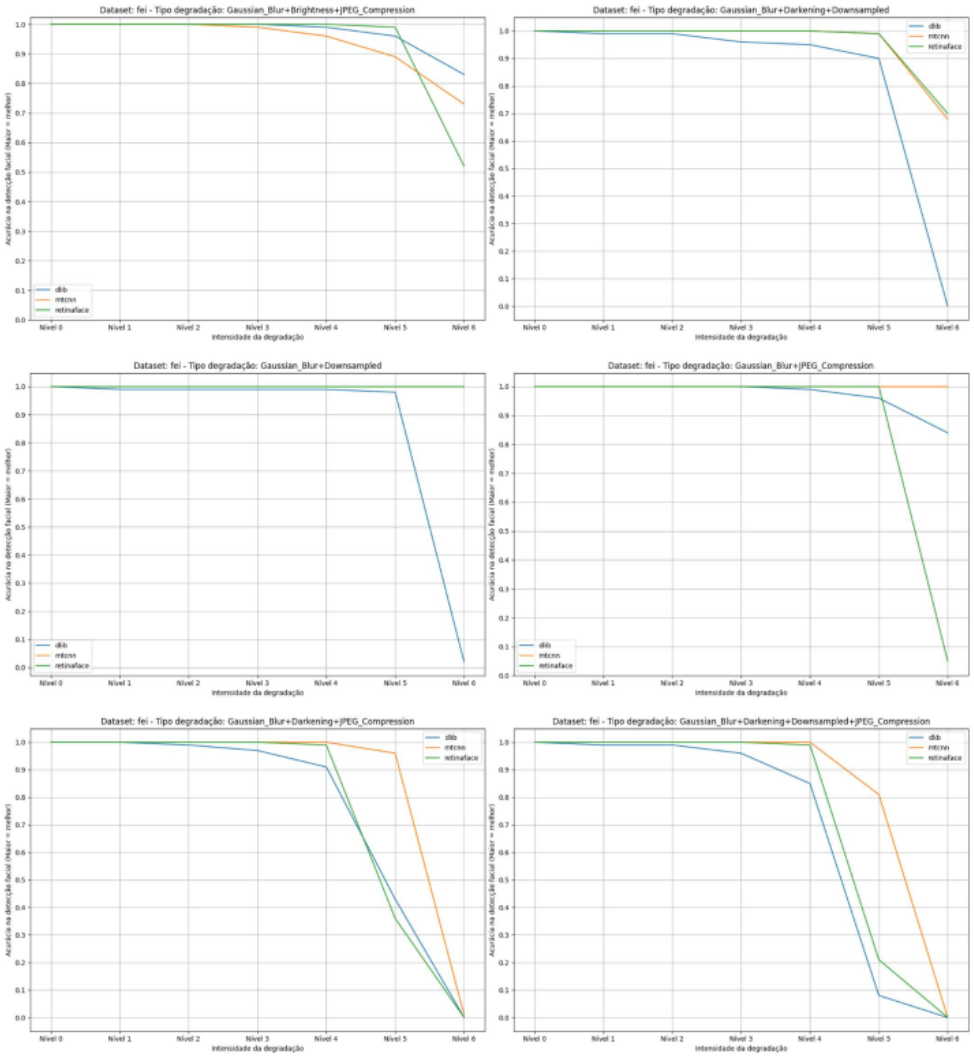


Figura I.100: Desempenho dos algoritmos de detecção facial (degradações em sequência) no *dataset* FEI

Dataset SCFace

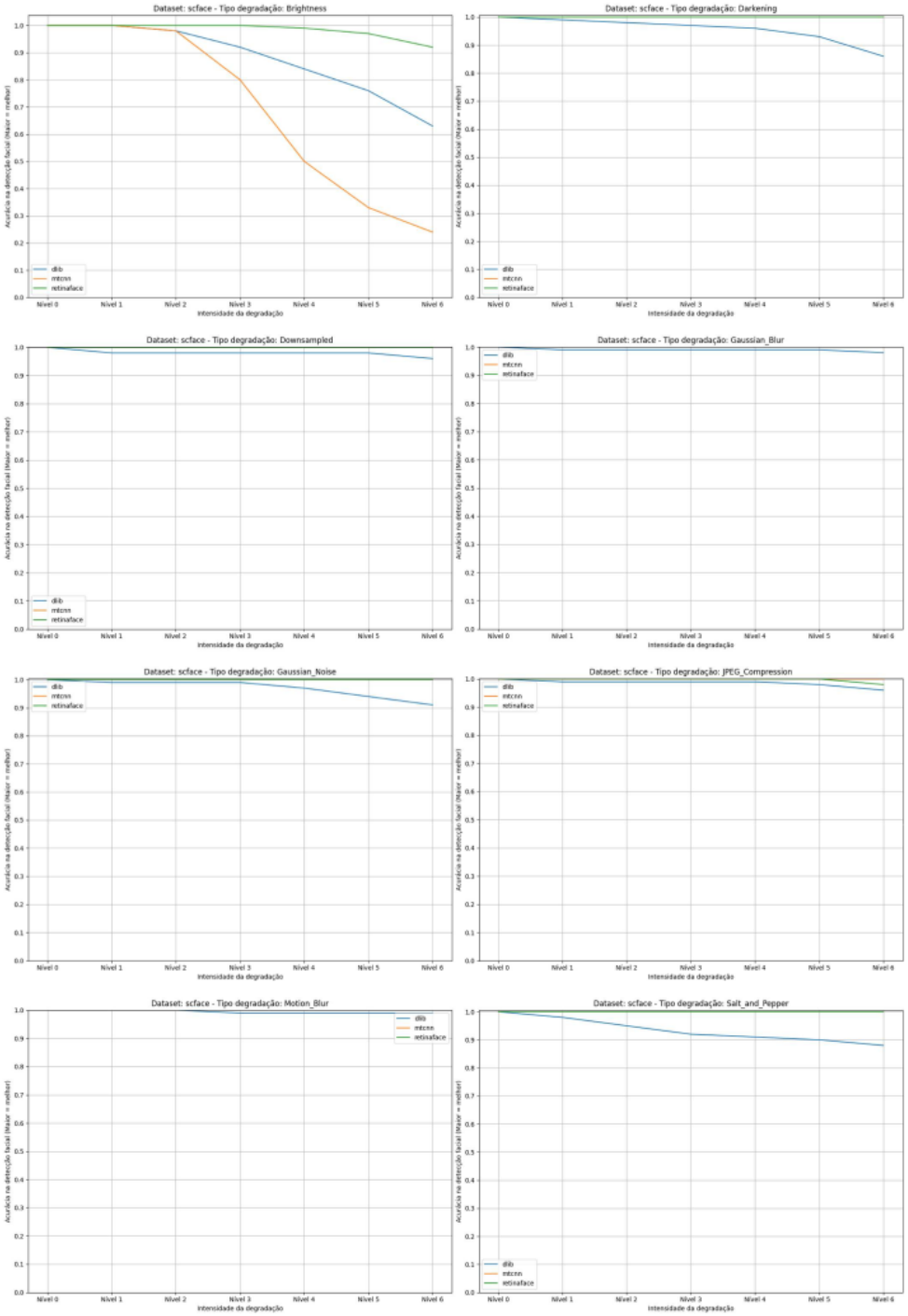


Figura I.101: Desempenho dos algoritmos de detecção facial (degradações simples) no *dataset* SCFace

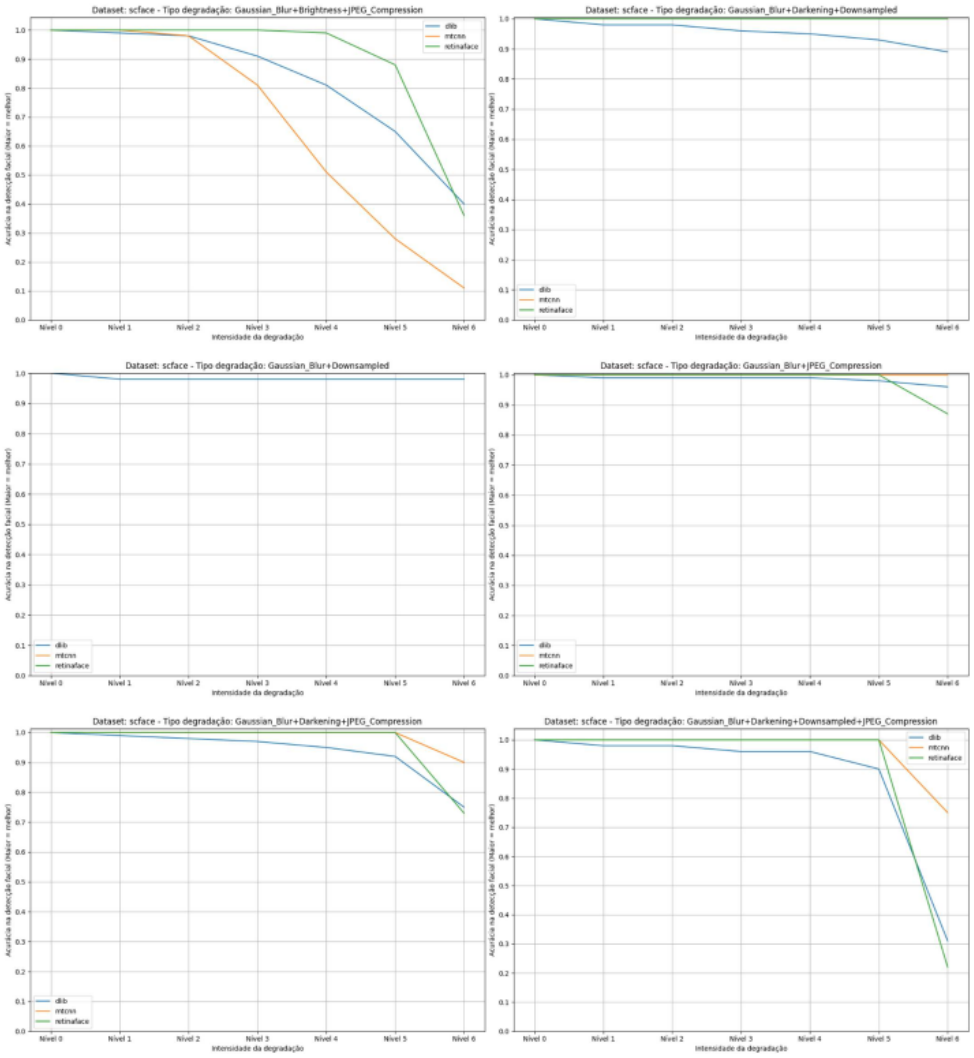


Figura I.102: Desempenho dos algoritmos de detecção facial (degradações em sequência) no *dataset* SCFace

Dataset GUFD

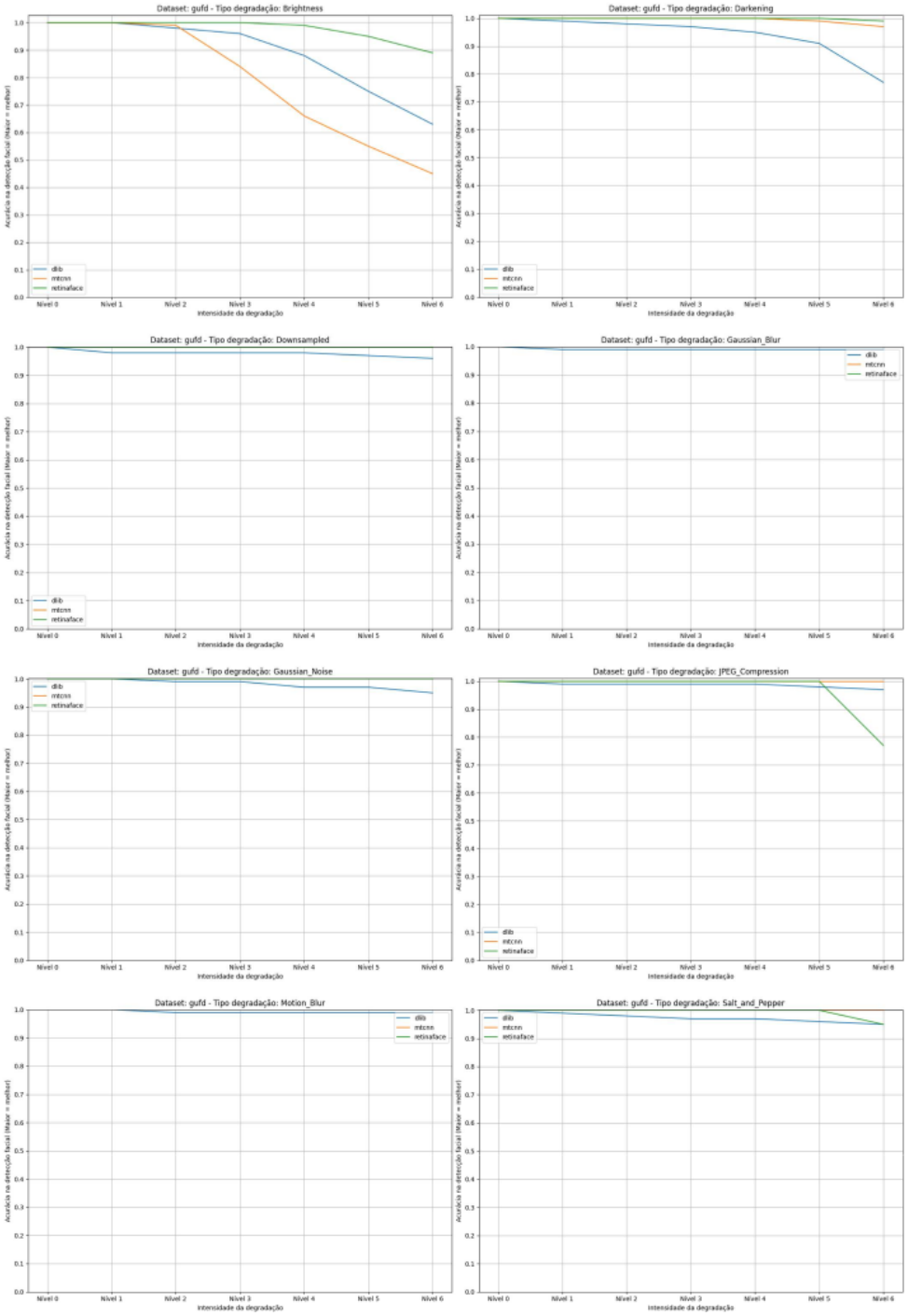


Figura I.103: Desempenho dos algoritmos de detecção facial (degradações simples) no *dataset* GUFD

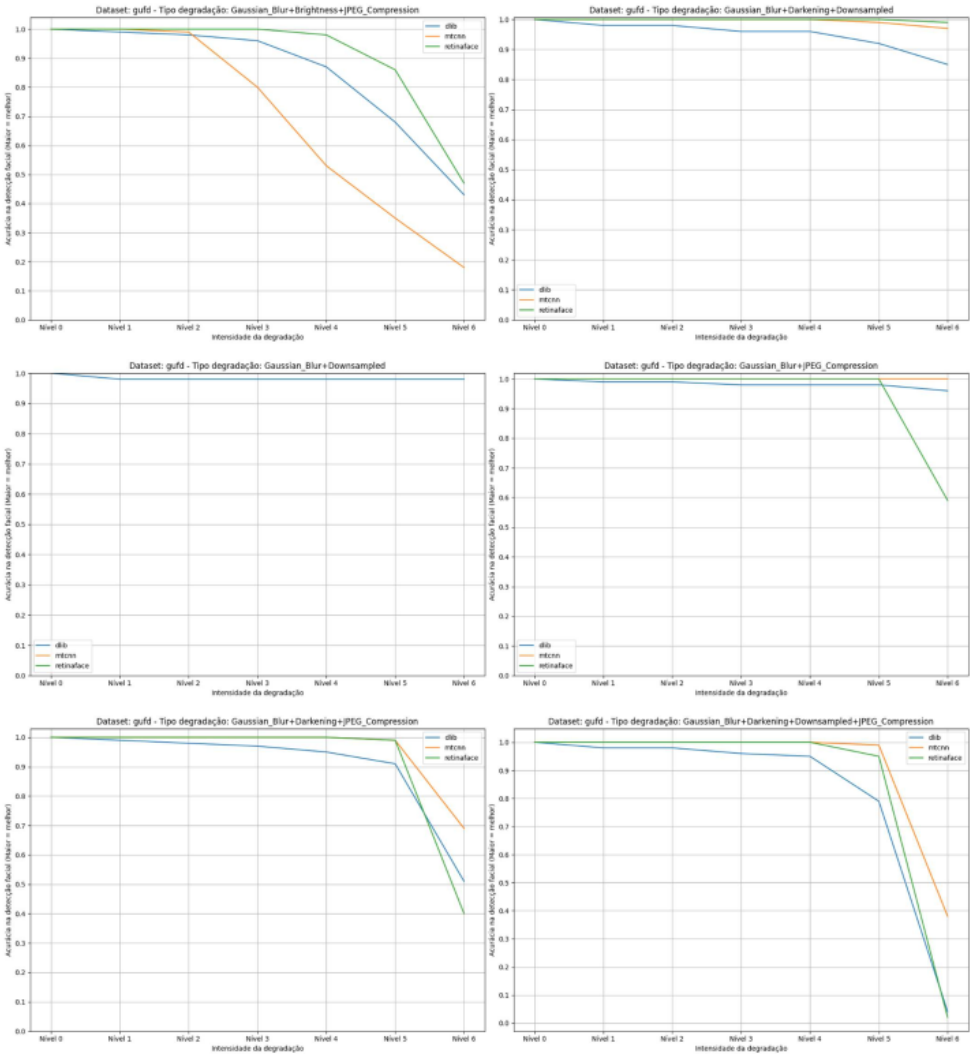


Figura I.104: Desempenho dos algoritmos de detecção facial (degradações em sequência) no *dataset* GUFD