

**MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
FARROUPILHA CAMPUS SANTO ÂNGELO**

**EXECUÇÃO DE TESTES DE RECONHECIMENTO AUTOMÁTICO DE DOR EM
RECÉM-NASCIDOS USANDO IMAGENS INTEIRAS E MARCAÇÕES FACIAIS NA
BASE ICOPE COM O MATLAB®**

TRABALHO DE CONCLUSÃO DE CURSO

WILLIAN FERST ROHTE

**Santo Ângelo, RS, Brasil,
2022**

WILLIAN FERST ROHTE

**EXECUÇÃO DE TESTES DE RECONHECIMENTO AUTOMÁTICO DE DOR EM
RECÉM-NASCIDOS USANDO IMAGENS INTEIRAS E MARCAÇÕES FACIAIS NA
BASE ICOPE COM O MATLAB®**

**Monografia apresentada ao Instituto Federal de Educação Ciência e Tecnologia
Farroupilha, como parte de requisito para obtenção do título de Tecnólogo em
Sistemas para Internet.**

Orientador: Prof. Dr. John Soldera

**Santo Ângelo, RS, Brasil,
2022**

**MINISTÉRIO DA EDUCAÇÃO SECRETARIA DE EDUCAÇÃO PROFISSIONAL E
TECNOLÓGICA INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E
TECNOLOGIA FARROUPILHA CAMPUS SANTO ÂNGELO**

**EXECUÇÃO DE TESTES DE RECONHECIMENTO AUTOMÁTICO DE DOR EM
RECÉM-NASCIDOS USANDO IMAGENS INTEIRAS E MARCAÇÕES FACIAIS NA
BASE ICOPE COM O MATLAB®**

Elaborado por

WILLIAN FERST ROHTE

Como requisito parcial para obtenção do título de

Tecnólogo em Sistema para Internet

COMISSÃO EXAMINADORA

John Soldera, Dr. (Orientador) (IFFar)

Talitha Comaru, Dr. (IFFar)

Fábio Weber Albiero, Me (IFFar)

Conceito Final: _____

Santo Ângelo, 22 de Setembro de 2022

DEDICATÓRIA

Dedico o presente trabalho de conclusão de curso aos meus pais que sempre me apoiaram na busca por conhecimento e me incentivaram a continuar e vencer as adversidades.

AGRADECIMENTOS

Sou grato primeiramente a Deus pelo dom da vida e sabedoria.

Aos meus pais por sempre me incentivarem e acreditarem que eu seria capaz de superar os obstáculos que a vida me apresentou.

Agradeço ao meu professor orientador que durante todas as etapas, esteve presente no projeto e me direcionou da melhor forma possível para conclusão do trabalho.

Aos professores do IFFar que me instruíram desde o início com um ensino de excelência, e me proporcionaram a evoluir continuamente.

A todos os colegas do curso de Sistemas para Internet, que diante de todo o empecilho no momento que estamos vivendo continuaram firmes, e se ajudando durante todo o caminho. E por fim, através da colaboração de todos as dificuldades desaparecem.

EPÍGRAFE

“A persistência é o caminho do êxito.”

(Charles Chaplin)

RESUMO

O presente trabalho visa a realizar testes de reconhecimento automático da dor em recém-nascidos, nos quais os valores dos parâmetros do método usado foram variados na execução do protótipo no Matlab® visando ao aumento das taxas médias de acerto, contribuindo assim com o projeto de “Pesquisa e aplicação de técnicas de reconhecimento facial”, também foram considerados 2 arquivos de testes, um que usa extração de feições a partir de marcações faciais e outro de imagens inteiras, estas imagens são de recém-nascidos entre 18 e 72 horas provenientes de uma base de imagens disponível na Internet. Foram realizados testes comparativos no protótipo do projeto entre diferentes métodos de extração de feições, redução de dimensionalidade e classificação. Os testes apontaram uma elevada taxa de acerto no reconhecimento automático da dor em recém-nascidos, possibilitando futuramente que seja desenvolvido um aplicativo para celulares voltado à área da medicina.

Palavras-chaves: medição da dor, recém-nascidos, imagens faciais, taxas de reconhecimento.

ABSTRACT

This work aims to perform automatic pain recognition tests in newborns, in which the parameter values of the method used were varied in the execution of the prototype in Matlab® aiming to increase the average success rates, thus contributing to the project "Research and application of facial recognition techniques", also considered 2 test files, one that uses feature extraction from facial markings and another of whole images, these images are of newborns between 18 and 72 hours from an image base available on the Internet. Comparative tests were performed on the project prototype between different methods of feature extraction, dimensionality reduction and classification. The tests showed a high rate of accuracy in the automatic recognition of pain in newborns, allowing the future development of an application for cell phones focused on the medical area.

Keywords: pain assessment, newborn, facial images, recognition rates.

SUMÁRIO

1 INTRODUÇÃO	12
1.1 Objetivos	13
1.1.1 Objetivo Geral	13
1.1.2 Objetivos Específicos	13
1.2 Fundamentação Teórica	13
1.2.1 Base iCOPE (infant Classification Of Pain Expressions)	14
1.2.2 Métodos de Reconhecimento Facial	14
1.2.2.1 Detecção de face - base de imagens	14
1.2.2.2 Normalização	15
1.2.2.3 Reconhecimento	15
1.2.3 Método PCA	16
1.2.4 Método Eigenface	17
1.2.5 Método Fisherfaces - LDA	17
1.2.6 Redução de dimensionalidade	19
1.2.7 Classificadores	19
1.2.7.1 Classificador do Vizinheiro Mais Próximo (NN)	19
1.2.7.2 Máquinas de Vetores de Suporte (SVM)	19
1.2.8 Matlab®	20
1.3 Trabalhos Correlatos	20
1.3.1 Avaliação de dor em Expressão Facial Neonatal por meio de Redes Neurais Profundas	20
1.3.2 Avaliação da dor neonatal a partir da Expressão Facial utilização de redes neurais profundas	21
1.3.3 Avaliação da Dor pela Expressão Facial: Neonatal Rede Neural Convolutacional (N-CNN)	22
2 DESENVOLVIMENTO	23
2.1 Caso de Uso	23
Figura 1: Diagrama de Caso de uso	23
2.1.1 Especificações do caso de uso	23
2.2 Diagrama de atividades	24
Figura 2: Diagrama de Atividades	25
Figura 3: Exemplo de imagens faciais da base iCOPE	26
Figura 4: Marcações Faciais na base iCOPE	26

Figura 5: Topologia adotada nas marcações	26
3 RESULTADOS E DISCUSSÕES	29
3.1 Testes Realizados	29
3.1.1 Ambiente de teste (configuração da máquina)	32
3.1.2 Quantidade de teste em cada arquivo	32
3.2 Comparação de classificadores SVM e NN	33
3.3 Comparação das técnicas de redução de dimensionalidade (LDA, PCA, AOLPP)	34
3.4 Influência dos valores dos parâmetros dos testes	35
Figura 6: Gráfico de um conjunto de resultados obtidos com a execução do arquivo de imagens inteiras	37
Figura 7: Gráfico de um conjunto de resultados obtidos com a execução do arquivo com marcações	38
4 CONSIDERAÇÕES FINAIS	39
REFERÊNCIAS	40
APÊNDICE A	43
APÊNDICE B	54
APÊNDICE C	65

1 INTRODUÇÃO

No seu livro A ORIGEM DAS ESPÉCIES, Charles Darwin (1809-1882), descreveu que, para a sobrevivência é essencial que os seres humanos e animais sintam emoções, desde o primeiro dia de vida quando nascem bebês de todas as espécies que sentem algum tipo de dor, incluindo a humana. A dor é fundamental para todo ser vivo, mantendo-o alerta contra reações psicológicas e fisiológicas que o levam a se proteger contra possíveis alterações no seu organismo (DARWIN, C., 1859).

A dor é definida como uma experiência sensorial e emocional desagradável, associada a uma lesão tecidual real (GALVAN, C., 2020). Com o avanço da medicina e estudos em recém-nascidos, os diagnósticos ficaram mais acessíveis em doenças neonatais. Essas doenças podem afetar toda a vida da pessoa caso não sejam descobertas no início da vida do neonato. Pensando nisso, um estudo denominado “Pesquisa e Aplicação de Técnicas de Reconhecimento Facial” está sendo realizado pelo orientador deste trabalho de conclusão de curso, Prof. John Soldera, do Instituto Federal Farroupilha - IFFar Câmpus de Santo Ângelo, com uma base denominada base iCOPE (infant Classification Of Pain Expressions) em tradução livre (Classificação infantil de expressões de dor) que está disponível na internet, para fins de estudos científicos, foi concedida a sua autorização pelo autor (BRANHAM, S. et. al, 2006) e neste trabalho foram utilizadas em torno de 204 imagens de crianças, nessa base há imagens classificadas como com dor e sem dor. (SOLDERA, J., et.al, 2021).

Este trabalho de conclusão de curso foi elaborado pensando em atribuir as maiores taxas de acerto possíveis para o estudo do grupo de pesquisa e tecnologia da informação em recursos terapêuticos e diagnósticos em saúde, seguindo a linha de pesquisa de reconhecimento facial e processamento de imagens, atuando na grande área de ciências exatas e da terra, subárea metodologia e técnicas da computação e na área da ciências da computação. Também foi usado um protótipo que foi adaptado para o reconhecimento de dor. E o protótipo usa um método de reconhecimento facial que possui as seguintes etapas: extração de feições de imagens faciais inteiras ou de marcações faciais, redução de dimensionalidade e por fim classificação. (SOLDERA, J, et al., 2015).

No apêndice C deste trabalho é apresentado um artigo que foi desenvolvido por (SOLDERA, J., 2021) para a Conferência Brasileira de Saúde em Informática - CBIS.

1.1 Objetivos

1.1.1 Objetivo Geral

Comparar resultados dos testes executados no protótipo desenvolvido na pesquisa mencionada na Introdução deste Trabalho de conclusão de curso, para auxiliar no projeto de reconhecimento de dor em recém-nascidos e conseqüentemente melhorar os resultados obtidos nos testes para eficácia dos algoritmos.

1.1.2 Objetivos Específicos

Executar testes diretamente no protótipo, com parâmetros variados e algoritmos de reconhecimento facial para verificação dos melhores resultados;

Apresentar os resultados com o uso do protótipo.

1.2 Fundamentação Teórica

Este trabalho de conclusão de curso, realiza a medição automática da dor através de técnicas de reconhecimento facial. Seu funcionamento será apresentado no decorrer deste trabalho de conclusão.

Durante o projeto de pesquisa denominado de “Reconhecimento Automático de Dor em Recém-Nascidos”, foi desenvolvido um protótipo de linguagem de programação científica no Matlab®, que realiza a avaliação automática da dor em recém-nascidos, fazendo uso de um método científico. Este protótipo foi desenvolvido pelos professores pesquisadores do projeto mencionado anteriormente, neste trabalho de conclusão, pretende-se executar testes com o protótipo a fim de se melhorar os resultados obtidos.

Para executar os testes, foram usados arquivos de teste que possuem a configuração de parâmetros para guiar o aprendizado de máquina realizado pelo protótipo. O protótipo foi desenvolvido pelos professores pesquisadores do projeto no software de linguagem de programação científica Matlab®, e o mesmo pode realizar cálculos numéricos simples até os mais avançados, e também foi utilizada a base iCOPE disponível na internet e explicada no item 1.2.1 deste trabalho.. O protótipo realiza a avaliação automática da dor em recém-nascidos com base nas configurações do arquivo de teste. Este trabalho tem o propósito

de executar testes tentando chegar aos melhores resultados possíveis, inclusive melhorar o aprendizado de máquina para estabelecer uma comparação entre diferentes formas de se realizar o aprendizado da máquina.

1.2.1 Base iCOPE (infant Classification Of Pain Expressions)

É uma base de imagens que foi desenvolvida por (BRAHNAM, S., 2006) durante um estudo no Hospital St. John (agora chamado Hospital da Misericórdia) com o Departamento de Neonatologia no estado de Missouri, EUA e consiste de 204 fotografias de 26 recém-nascidos experimentando diferentes estímulos nocivos, incluindo repouso e estímulo de dor aguda. Para fins científicos foi concedida autorização do seu uso pelo autor. Todas as imagens foram fotografadas com uma câmera digital Nikon D100 em condições de luz ambiente com uma resolução de 3008x2000 numa sala separada de outros recém-nascidos.

As imagens pré-processadas da base foram subdivididas aleatoriamente em dois grupos: aprendizado de máquina e validação. (BRAHNAM S. et. al, 2006.).

Durante os testes, foram analisadas um total de 204 fotografias coloridas de 26 recém-nascidos, sendo 13 do sexo masculino e 13 do sexo feminino, com idade variando entre 18 e 72 horas. Estas imagens são divididas da seguinte forma: 144 são de recém-nascidos sem dor e outras 66 imagens são com neonatos com dor.

1.2.2 Métodos de Reconhecimento Facial

Para reconhecer uma face específica em um banco de imagens contendo diversas faces, divide-se este processo em três fases de maneira simplificada: detecção, normalização e reconhecimento de faces. Nestas fases podem-se utilizar diversas técnicas, como com o uso de redes neurais artificiais, análise dos componentes principais (PCA).

1.2.2.1 Detecção de face - base de imagens

Esta é a primeira etapa do processo de reconhecimento de face, que se resume em localizar uma face em uma imagem que normalmente possui muitos elementos complexos que dificultam o posicionamento de uma face na imagem, para isto podemos utilizar diversas técnicas, como o método “Viola-Jones”.

A detecção facial consiste em identificar e isolar a área referente à face em uma imagem digital e pode ser realizada com base em vários atributos: formato do rosto ou cabeça, aparência da face, ou a combinação destes.

1.2.2.2 Normalização

Esta fase tem a função de padronizar a face para que se possa seguir o mesmo tratamento dado às faces existentes no banco de dados, alterando, por exemplo, as características contidas na imagem obtida, tais como alteração do padrão de cor da imagem, seu tamanho e outras características para se assimilarem com as restantes no banco.

Na etapa de extração de características o objetivo é localizar regiões da imagem que contenham características significativas podendo essas características ser distinguidas por suas texturas, formas, intensidade entre outros. Ao contrário da detecção facial que busca padronizar a forma da face para assim detectá-la.

Neste projeto de reconhecimento facial, as imagens utilizadas da base Icope já foram detectadas e normalizadas, portanto, estes passos não precisaram ser realizados.

1.2.2.3 Reconhecimento

Após a fase de detecção e normalização contemplamos a fase de reconhecimento de face propriamente dito. Nesta fase avalia-se a imagem a ser reconhecida, comparando-a com as que estão em um conjunto armazenado no banco, através dos cálculos necessários para efetuarmos essa comparação, acessando-as de maneira mais eficiente possível, ou seja, diminuindo a porcentagem de erro. Para métodos que reduzem o espaço de características, como o PCA, torna-se necessário a utilização de um classificador, que tem como função, determinar valores numéricos singulares para a comparação, verificando os valores que mais se aproximam.

A etapa de reconhecimento facial consiste em encontrar em meio a um conjunto pré-definido de faces àquela que mais se aproxima da face que está sendo analisada através da análise e comparação das características extraídas na etapa de extração.

1.2.3 Método PCA

A PCA é uma técnica matemática que descreve um conjunto de dados usando os principais componentes que representam da melhor maneira o conjunto de dados, usados de maneira a reduzir a dimensionalidade dos dados ou então detecção de padrões. Quando aplica-se a transformada de Fourier, o resultado é a mudança de base, uma projeção em um novo espaço onde cada componente seja expressa em ordem de variância ou em contribuição ao conjunto de dados. (PENHARBEL et al., 2005).

Um método muito difundido no tratamento de características é o de Análise das Componentes Principais (PCA - Principal Component Analysis), por ser o primeiro método de reconhecimento facial a obter sucesso e ainda hoje é muito usado para comparação a novas propostas de reconhecimento facial.

A técnica PCA é descrita no algoritmo a seguir:

A) Matriz D: Organize os dados em uma matriz D, onde a primeira linha será formada pelas componentes da primeira amostra, a segunda linha pelas componentes da segunda amostra e proceda até a N-ésima linha que será formada pela N-ésima amostra com suas componentes.

B) Vetor Média: Cria-se um vetor-média E, formado pelas médias de cada coluna M da matriz D.

C) Subtração da média: Subtrair de cada item de cada coluna M da matriz D a média calculada em E correspondente a qual o item pertença.

D) Matriz de Covariância: Calcule a matriz de covariância da matriz D.

E) Autovetores e Autovalores: Da matriz de covariância D, calcula-se os autovetores e autovalores gerando uma matriz de autovetores e um vetor de autovalores.

F) Ordenação dos valores: Ordena-se a matriz de autovetores pela ordem crescente de autovalores do vetor de autovalores correspondentes.

1.2.4 Método Eigenface

O método Eigenface baseia-se em linearmente, projetar o espaço de imagens em um espaço de características com dimensões reduzidas obtido fazendo uso da análise de componentes principais (PCA), também conhecido como método Karhunen-Loeve. Entretanto, produz direções de projeção que maximizam a dispersão dos pontos no gráfico em todas as classes, isto é, em todas as imagens faciais mantém as variações indesejadas causadas pela iluminação e expressão facial.

O método Eigenfaces busca por um conjunto de características que não dependem somente dos atributos geométricos da face (distância entre olhos, nariz e boca), mas sim de toda a informação da representação facial. O Eigenfaces identifica as características mais relevantes que diferenciam uma face específica dentre outras, criando uma imagem para cada característica.

Baseadas na Teoria da Informação, as eigenfaces buscam identificar um baixo número de características, mas que são relevantes para diferenciar uma face de outras faces. Essas características podem ser analisadas apenas com a variação dos valores assumidos pelos pixels, em um conjunto de imagens de faces.

1.2.5 Método Fisherfaces - LDA

Com o mesmo objetivo da técnica PCA, temos a LDA que consiste em reduzir a dimensionalidade dos dados visando a classificação. A partir de um conjunto de dados multidimensionais rotulados, o LDA gera um conjunto de dados de menor dimensionalidade que representa as classes dos dados originais.

Belhumeur et al (1997) propôs o método Fisherfaces que classifica as faces em conjuntos ou classe pois trabalha com o uso de “rótulos”, isto é, uma vez identificado os rostos dizendo qual face pertence a qual pessoa, os mesmos são agrupados por pessoa, e cada agrupamento desses é conhecido como uma classe. O método tenta modelar a dispersão dos pontos visando maior confiabilidade para a classificação.

Similar com as eigenfaces, as Fisherfaces podem ser visualizadas como imagens de características onde as características das Fisherfaces são variações de aparência presentes nas

imagens de cada indivíduo, tais como variações de luminosidade, poses e expressões faciais. Assim como as imagens no espaço de dados possuem um valor para cada atributo, os vetores e características possuem um valor para cada Fisherface (faces fisher).

Análise Discriminante Linear (LDA): Esta técnica realiza a redução por meio de um mapeamento dos objetos de alta dimensionalidade para um espaço de características de menor dimensionalidade. Para realizar o mapeamento, a LDA cria um vetor E para construir um espaço de características que busca minimizar a dispersão intra-classe e maximizar a dispersão interclasse, ou seja, aproximar os objetos de uma mesma classe e distanciar objetos de classes distintas.

Esta técnica realiza a redução por meio de um mapeamento dos objetos de alta dimensionalidade para um espaço de características de menor dimensionalidade. Para realizar o mapeamento, a LDA cria um vetor E para construir um espaço de características que busca minimizar a dispersão intra-classe e maximizar a dispersão interclasse, ou seja, aproximar os objetos de uma mesma classe e distanciar objetos de classes distintas.

Laplacianfaces (LPP): – Aplica primeiramente PCA para compactar os dados faciais. Cria-se um grafo para preservar a estrutura local dos dados (HE et al. 2005).

OLPP (Orthogonal Laplacianfaces): Estende o método Laplacianfaces de forma a obter uma projeção ortogonal (CAI, et al 2006).

COLPP/AOLPP: Extensão do OLPP de forma a preservar também a estrutura das classes no espaço de baixa dimensionalidade. (Soldera, J, et al., 2015).

O método Projeções Ortogonais Preservadoras de Localidade (Orthogonal Locality Preserving Projections - OLPP) (CAI et al., 2006) foi proposto para melhorar a preservação da estrutura local dos dados faciais e, conseqüentemente, incrementar a discriminação das classes de dados faciais no espaço de baixa dimensionalidade. Em outras palavras, o método OLPP adiciona a propriedade de ortogonalidade à matriz de transformação obtida pelo método LPP causando uma melhor preservação da métrica de classificação facial no espaço de baixa dimensionalidade.

1.2.6 Redução de dimensionalidade

Métodos de redução de dimensionalidade visam diminuir o número de dimensões dos vetores originais, mapeando-os para um novo espaço.

Técnicas lineares mapeiam os vetores originais para espaços que são compostos por combinações lineares dos vetores originais, ou seja, multiplica cada vetor original por uma matriz de transformação. Métodos não-lineares usam transformações não lineares para reduzir o número de eixos.

Então, a dimensionalidade dos vetores é reduzida usando o método COLPP, levando a um número de dimensões escolhido que é reduzido e, por fim, submetidas ao classificador. (Soldara, J. et al., 2015).

1.2.7 Classificadores

1.2.7.1 Classificador do Vizinho Mais Próximo (NN)

Determina a amostra de treinamento mais próxima da amostra de teste no espaço de baixa dimensionalidade. Os vizinhos mais próximos são definidos em termos da distância Euclidiana.

A medida da distância euclidiana é baseada na distância entre dois vetores representados num espaço multidimensional. Ela calcula a distância euclidiana entre dois pontos nesse espaço multidimensional. Seu cálculo é baseado no teorema de Pitágoras. (COVER; HART, 1967).

1.2.7.2 Máquinas de Vetores de Suporte (SVM)

Aqui é determinado qual é o melhor plano para seleção de duas classes. Utiliza margens suaves para realizar a classificação de novas imagens faciais. Uma SVM é um conceito que elabora um conjunto de métodos de aprendizado supervisionado. Este classificador pega um conjunto de dados e antecipa para qual entrada é concedida. Duas possíveis classes a entrada faz parte, o que faz do SVM um classificador binário não probabilístico. (VAPKINK, 1998).

Em outras palavras, o que uma SVM faz é encontrar uma linha de separação, chamada de hiperplano, entre dados de duas classes. Essa linha busca maximizar a distância entre os pontos mais próximos em relação a uma das classes.

1.2.8 Matlab®

É um software voltado para realização de cálculos numéricos com matrizes, processamento de sinais e construção de gráficos, O Matlab® é um sistema interativo cujo elemento básico de informação é uma matriz que não requer dimensionamento. Esse sistema permite a resolução de muitos problemas numéricos em apenas uma fração do tempo que se gastaria para escrever um programa semelhante em linguagem Fortran, Basic ou C. Além disso, as soluções dos problemas são expressas quase exatamente como elas são escritas matematicamente (MATLAB®, 2022).

1.3 Trabalhos Correlatos

1.3.1 Avaliação de dor em Expressão Facial Neonatal por meio de Redes Neurais Profundas

A avaliação da dor neonatal pode sofrer variações entre profissionais de saúde, resultando em intervenção tardia e tratamento inconsistente da dor. Neste contexto, o objetivo desta dissertação denominada “Avaliação de dor em Expressão Facial Neonatal por meio de Redes Neurais Profundas” (BUZUTI, L, THOMAS, C. E., 2020) foi analisar, quantitativa e qualitativamente, modelos de Redes Neurais Convolucionais na tarefa de classificação automática da dor neonatal por meio de imagens de faces de dois bancos de dados distintos (um internacional, denominado iCOPE (infant Classification Of Pain Expression), e outro nacional, denominado UNIFESP (Universidade Federal de São Paulo)). Os resultados quantitativos mostraram a superioridade da arquitetura N-CNN (*Neonatal Convolutional Neural Network*) traduzindo para o português (Rede Neural Convolucional Neonatal) para avaliação automática da dor neonatal, com taxas de acerto médias de 87.2% e 78.7% para os bancos de imagens iCOPE (infant Classification Of Pain Expressions) em tradução livre (Classificação infantil de expressões de dor) e UNIFESP (Universidade Federal do Estado de São Paulo), respectivamente. No entanto, a análise qualitativa evidenciou que todos os modelos neurais avaliados, incluindo a arquitetura Neonatal Convolutional Neural Network (N-CNN), podem aprender artefatos da imagem e nas variações discriminantes das faces,

mostrando a necessidade de mais estudos para aplicação de tais modelos na prática clínica em questão.

Foram escolhidos apenas os *ground truths* (imagens rotuladas como de dor e não-dor) para o conjunto de dados iCOPE, foram utilizadas 153 imagens coloridas com de 26 recém-nascidos caucasianos. Para a UNIFESP foram utilizadas 356 imagens coloridas de 30 recém-nascidos.

A taxa de acerto de treinamento, validação e teste de cada modelo treinado nos bancos de imagens UNIFESP e iCOPE. Verifica-se que em ambos os bancos de imagens o modelo N-CNN (Redes Neurais Convolucionais Neonatais) teve o melhor desempenho estatisticamente, 87.2% (iCOPE) e 78.7% (UNIFESP). (BUZUTI, L, THOMAS, C. E., 2020).

1.3.2 Avaliação da dor neonatal a partir da Expressão Facial utilização de redes neurais profundas

Este trabalho apresentado para o departamento de pediatria da Unifesp tem por objetivo investigar dois modelos de Aprendizado Profundo (DL), *Neonatal Convolutional Neural Network* (N-CNN) e ResNet 50, é uma rede neural convolucional treinada com diversas imagens retiradas da internet em bases de imagens distintas, e também pode ser utilizada para avaliação facial automática da dor neonatal. ResNet 50, possui 50 camadas de profundidade pré-treinadas a partir de mais de milhão de imagens do banco de dados ImageNet (*ImageNet*, 2022). *ImageNet*. Foi elaborado utilizando a linguagem de programação *Python*, utilizando 356 imagens da base da Unifesp. Nele foram explorados os métodos K-CNN (*Neonatal Convolutional Neural Network*) proposta por Zamzmi sendo que a mesma propôs pela primeira vez o estudo deste método em 2019. E outro método utilizado neste artigo, foi o ResNet 50 modificada por Zamzmi em 2019 através do conceito de Aprendizado por Transferência (TL).

Para a construção deste sistema computacional, foi utilizada a linguagem de programação *Python* em sua versão 3 e o *Open Source AI Framework*, para o aprendizado de máquina de alta performance da própria Google, denominada *Tensor Flow*. Bancos de imagens para incentivar estudos utilizando visão computacional, aprendizado de máquina e principalmente, aprendizado profundo na análise e avaliação de dor neonatal ainda não estão disponíveis em abundância para realização deste tipo de estudo.

Os resultados deste trabalho ficaram em 80% no conjunto de treinamento e 20% no conjunto de validação. Utilizando um método que não foi utilizado neste trabalho o Resnet 50 e uma base de imagens própria da Unifesp, alcançaram uma taxa de 87,50%. (BUZUTI, L. 2021).

1.3.3 Avaliação da Dor pela Expressão Facial: Neonatal Rede Neural Convolutacional (N-CNN)

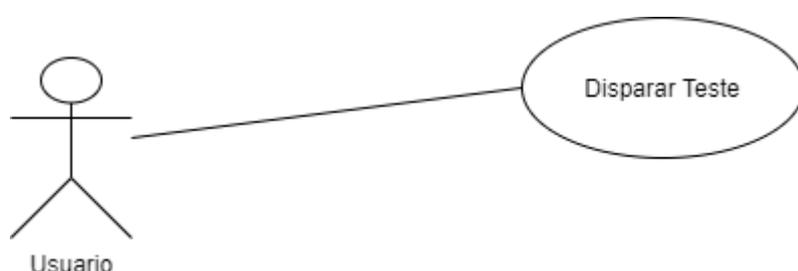
O padrão atual para avaliar a dor neonatal é descontínuo e sofre de variações interobservadores, que podem resultar em intervenção atrasada e tratamento incoerente de dor. Por conseguinte, é fundamental abordar as insuficiências do padrão atual e desenvolver contínuo e menos subjetivo instrumentos de avaliação da dor. As redes neurais convolucionais têm ganhado muita popularidade nas últimas décadas, devido à ampla gama das suas aplicações bem sucedidas na análise de imagem médica, reconhecimento de objetos, e reconhecimento de emoções. Neste trabalho denominado “Avaliação da Dor pela Expressão Facial: Neonatal Rede Neural Convolutacional (N-CNN)” é proposto uma Rede Neural Convolutacional Neonatal, concebida e treinada de ponta a ponta para detectar a dor neonatal. Também é avaliado uma rede proposta em dois conjuntos de dados de neonatos e comparada o seu desempenho para o desempenho da arquitectura ResNet em os mesmos conjuntos de dados. O método proposto por (ZAMZINI, G., 2019), superou a ResNet no reconhecimento da dor dos recém-nascidos e atingiu cerca de 84,5% de precisão utilizando a mesma base da iCOPE. (ZAMZINI, G., 2019).

2 DESENVOLVIMENTO

2.1 Caso de Uso

Este diagrama descreve as funcionalidades do sistema na visão do usuário final. Na figura 1 é ilustrado o único caso de uso para este trabalho. Os atores são os responsáveis pela interação do sistema com o usuário, e cada caso de uso significa uma funcionalidade disponível no sistema para este ator. Também para cada diagrama é apresentada uma descrição detalhada das funcionalidades do sistema.

Figura 1: Diagrama de Caso de uso



Fonte: Elaborado pelo Autor, 2021

2.1.1 Especificações do caso de uso

Descreve como serão executadas as ações do usuário no sistema, gerando assim uma resposta para o mesmo. Cada tarefa que o usuário realiza é executada uma sequência de ações que detalha como o sistema deve se comportar diante do usuário.

Abaixo, a tabela 1 apresenta as “Especificações do caso de uso - Disparar Teste”.

Tabela 1: Especificações do caso de uso “Disparar teste”

Nome do caso de uso	Disparar teste
Ator Principal	Usuário
Ator Secundário	(Não se aplica)
Pré-Condições	Ter o arquivo de teste com os parâmetros do teste

Pós-Condições	
Fluxo Principal	
Ações do Ator	Ações do Sistema
1 Abre o software para execução dos testes	
2 Executar o arquivo de teste	
	3 Executa o teste até o final ou até que o usuário cancele
	4. Exibe na tela a taxa de acerto média em todos os testes

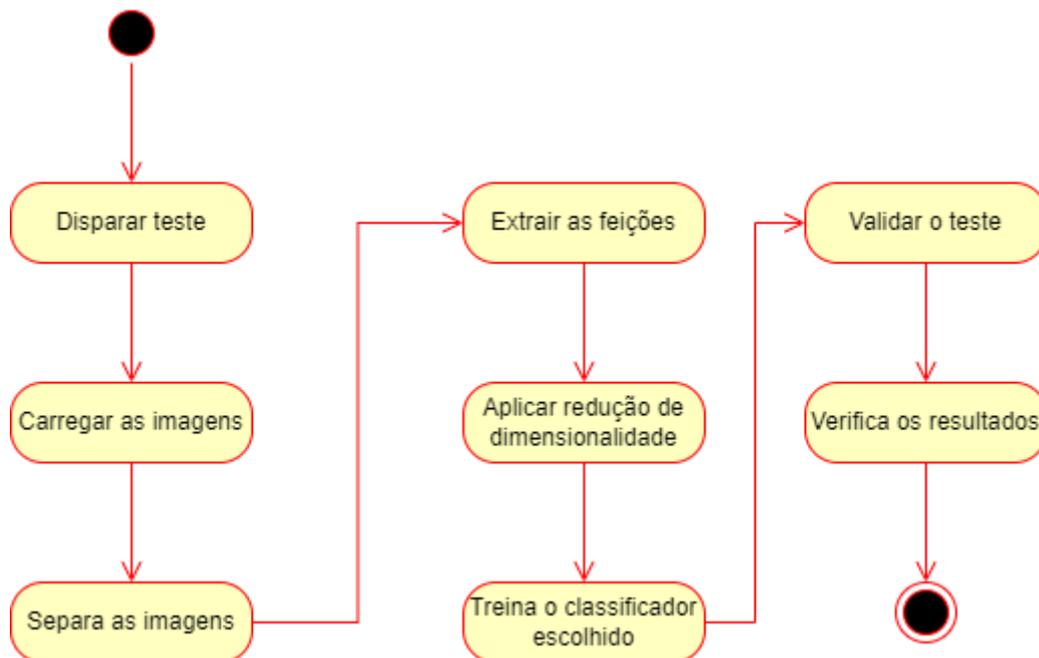
Fonte: Elaborado pelo Autor, 2021

2.2 Diagrama de atividades

O Diagrama de atividade é um diagrama definido pela Linguagem de Modelagem Unificada (UML), e representa os fluxos conduzidos por processos. É essencialmente um gráfico de fluxo, mostrando o fluxo de controle de uma atividade para outra. Comumente isso envolve a modelagem das etapas sequenciais em um processo computacional.

Na Figura 2, é apresentado o diagrama de atividades do caso de uso “Disparar Teste”.

Figura 2: Diagrama de Atividades



Fonte: Elaborado pelo Autor, 2021

A seguir são detalhados os passos do diagrama de atividades:

1º) Disparar teste: Com o software aberto o usuário deve disparar o teste com o protótipo (método experimental de reconhecimento facial) e aguardar sua finalização;

2º) Carregar as imagens: O protótipo carrega as imagens no Matlab® para que os testes possam ser executados;

Na Figura 5, logo abaixo, aplica-se a interpolação, ou seja, nos pares de marcações faciais são interpolados 5 pontos, igualmente espaçados (em cores diferentes). vermelho: pontos marcados pelos bolsistas do projeto do orientador e em azul, interpolados igualmente espaçados entre cada par de pontos de marcações consecutivos.

Os *landmarks* (marcadores faciais), são os pontos em vermelho da imagem abaixo, são marcações realizadas pelo autor deste trabalho de conclusão de curso, utilizando o software Matlab®.

Na figura 3, temos os modelos que são relacionados a base iCOPE, de dois estímulos, repouso, e dor aguda.

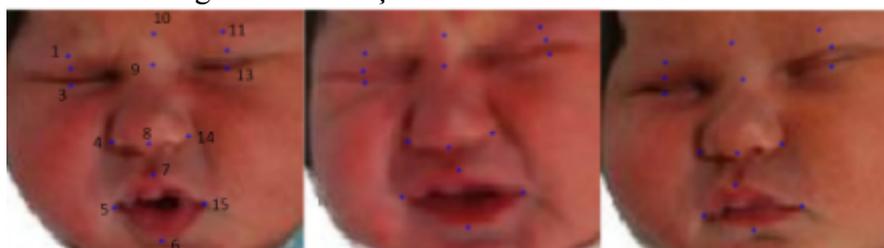
Figura 3: Exemplo de imagens faciais da base iCOPE



Fonte: Base iCOPE, 2021

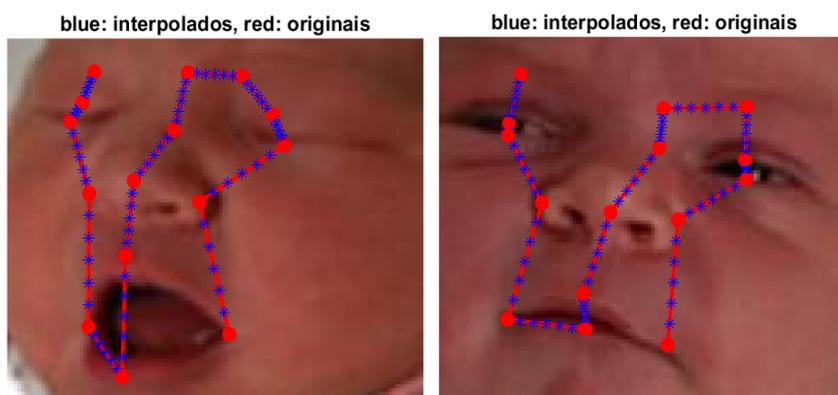
Na figura 4, são mostrados os pontos de marcações faciais em ambos os casos, com dor e sem dor.

Figura 4: Marcações Faciais na base iCOPE



Fonte: Base iCOPE, 2021

Figura 5: Topologia adotada nas marcações



Fonte: Base iCOPE, 2021

3º) Separar as imagens: O sistema automaticamente separa uma determinada quantidade de imagens para treinamento e outra quantidade para o teste;

4º) Extrair as feições: O sistema usa imagens inteiras ou as imagens com marcações faciais para realização dos testes, inicialmente cada imagem RGB foi reduzida ao tamanho de 64x48 pixels e transformada em um vetor de feições concatenando todos os seus pixels. Nesta etapa, pode-se usar imagens inteiras ou marcações, no caso de imagens inteiras, elas são reduzidas de tamanho para reduzir a complexidade computacional e nas marcações, também foram feitas marcações de pontos nas imagens dos recém nascidos, seguindo partes mais sensíveis a dores, que seriam basicamente, em torno da sobrancelha, com um ponto no início e no fim de cada uma, nos cantos dos olhos, em cima do nariz, contornando a boca e dois pontos que definem as bochechas. Essas marcações puderam melhorar a taxa de acerto da classificação da dor em recém nascidos através da utilização de software. Foram marcadas para estes testes 204 imagens.

Os pontos faciais abaixo descritos foram marcados para extrair vetores de feições.

1: Acima da sobrancelha esquerda,

2: Acima do olho esquerdo,

3: Abaixo do olho esquerdo,

4: Canto esquerdo do nariz,

5: Canto esquerdo da boca,

6: Parte inferior do lábio,

7: Parte superior do lábio,

8: Parte inferior do nariz,

9: Meio dos olhos,

10: Meio das sobrancelhas,

11: Acima da sobrancelha direita,

12: Acima do olho direito,

13: Abaixo do olho direito,

14: Canto direito do nariz,

15: Canto direito da boca.

Para que se pudessem ser realizados os testes de reconhecimento automático da dor em recém-nascidos deste trabalho foi usado um protótipo desenvolvido no Matlab® por (Soldara et al 2015.)^[12] que usa técnicas de reconhecimento facial adaptadas ao reconhecimento de dor em recém-nascidos. no Matlab®, que se destina a cálculos numéricos e gráficos científicos. O destaque desta linguagem está na manipulação e cálculos matriciais, como por exemplo, resolução de sistemas lineares, cálculo de autovalores e autovetores, fatora, e função de matrizes, entre outros. Além disso, esse software possui diversas funções que auxiliam no desenvolvimento do protótipo.

5º) Aplica uma técnica de redução de dimensionalidade adequada de forma a mapear os vetores de feições do espaço original de alta dimensionalidade para um espaço de dimensões reduzidas, onde os vetores resultantes podem ser mais eficientemente manipulados.

6º) Treinar o classificador: Aqui o sistema treina o classificador escolhido o vizinho mais próximo (NN) ou máquina de vetores e suporte (SVM).

7º Validar o teste: O usuário valida o teste se os resultados estão de acordo com o esperado, caso contrário, pode-se alterar algumas variáveis para melhorar as taxas de acerto.

8º Verificar os resultados: O usuário verifica os resultados se estão de acordo com os testes então valida os resultados e inclui na execução do projeto.

3 RESULTADOS E DISCUSSÕES

3.1 Testes Realizados

Os arquivos de teste usam um protótipo de reconhecimento facial elaborado por (SOLDERA, J. et. al. 2015), utilizando o Matlab®, explicados na seção anterior, sendo assim, este trabalho de conclusão de curso tem por objetivo apresentar os melhores resultados possíveis, para que seja possível se obter taxas de acerto mais elevadas, auxiliando no projeto de reconhecimento facial mencionado anteriormente.

São arquivos que configuram um plano de testes sobre uma base de imagens, mais especificamente, os arquivos de testes informam os parâmetros usados pelo protótipo. Os parâmetros são subdivididos em:

- Configurações da base de imagens:

Neste projeto, foi utilizada uma base disponível na internet chamada de iCOPE, nos arquivos de testes para alcançar a taxa de acerto têm-se vários parâmetros onde se altera os valores e roda novamente os testes para definição.

As taxas de acerto apresentadas posteriormente neste projeto, definem-se como a média das vezes que o classificador atribui a uma amostra de teste à classe correta dado o conjunto total de experimentos (10 iterações vezes 10 classificações do ten-fold, ou seja, a média do acerto em 100 experimentos para cada imagem). Foram testados um número maior de iterações, porém os resultados mantiveram-se praticamente os mesmos, por outro lado, foram executados testes com número menor de iterações tendo neste caso variabilidade da taxa média, portanto, 10 iterações se mostrou com melhor equilíbrio entre tempo de execução e variação da taxa média de acertos.

Os valores das variáveis variam a cada teste realizado, pois o mesmo interfere na taxa de acerto, aqui neste projeto são apresentados as variáveis que nestas configurações atingiram a maior taxa possível.

- Parâmetros de Organização dos testes:

crossValidation: Quando é definido como “true” usa validação cruzada, e quando é definido como “false” usa o método de teste leave-one-out.

kFold: é o número de repetições quando usada validação cruzada (k-fold), nos testes é definido como 10 (10-fold).

O método de validação cruzada denominado *k-fold* consiste em dividir o conjunto total de dados em k subconjuntos mutuamente exclusivos do mesmo tamanho e, então, um subconjunto é utilizado para teste e os $k-1$ restantes são utilizados para estimação dos parâmetros, Calculando a taxa de acerto do modelo. Este processo é repetido k vezes alternando de forma circular o subconjunto de teste.

Esta metodologia é usada em ambos os arquivos de teste, os experimentos repetidos e a taxa de acerto obtida é a taxa média das 10 repetições. k-fold (ou seja, 10-fold) que é um tipo de validação cruzada. Separa todas as amostras em 10 conjuntos, e seleciona 1 para teste e os outros para treinamento, repete variando o conjunto de teste até que todos tenham sido testados.

- Parâmetros de Repetição dos testes:

iterations: Número de iterações do mesmo teste, ou seja, o número de vezes em que tudo é repetido novamente incluindo a seleção aleatória das imagens (nos testes esse parâmetro foi fixado em 10).

- Parâmetros para seleção de feições

colorChannels: feições escolhidas, no caso canais de Cor (vermelho, verde e azul).

resizeImage: true, redimensiona as imagens: true: muda o tamanho e false: mantém o tamanho original (com marcadores faciais, as imagens não são redimensionadas).

newImageSize: [64, 48], novo tamanho de imagem se selecionado parâmetro para redimensionar.

- Parâmetros de pós-processamento de feições

method: é o método de escolha de feições, 1: imagens inteiras que são convertidas para vetores, 3: concatena as vizinhanças dos marcadores faciais para formar vetores de feições.

- Parâmetros de Redução de Dimensionalidade

method: é o método de redução de dimensionalidade: 0=nenhum, 1=PCA, 2=LDA , 7=AOLPP.

PCARatio: é a taxa de compactação das feições após a aplicação da técnica PCA: usado nos métodos PCA, LDA ou AOLPP.

ReducedDim (Redução de Dimensionalidade): Número de dimensões do vetor resultante após aplicação da redução de dimensionalidade escolhida.

K: É o número de imagens vizinhas para determinar a vizinhança dos dados, ou seja, a estrutura dos dados (os melhores resultados são com k igual ao o número de imagens de treinamento menos 1).

supervised (verdadeiro ou falso): quando esse parâmetro é verdadeiro é usada a informação de a qual classe pertencem as imagens no treinamento (dor ou não-dor) levando a uma melhor preservação das classes ao invés das estrutura dos dados. E quando é falso, se refere à abordagem não-supervisionada, portanto, no treinamento, não considera as classes da imagem, levando à preservação da estrutura dos dados e conseqüentemente a pior separabilidade de classes.

- Parâmetros de Classificação (os mais importantes)

method: 1; é o método de classificação escolhido para realização do teste;

embeddSVM: false, 1: NN (nearest neighbor), 4: SVM soft margins neste projeto foram utilizados apenas os classificadores NN e SVM nos testes.

SVM.soft: true, se o SVM usa margens suaves (se para determinar a fronteira dados atípicos são minimizados).

Os arquivos de teste utilizados neste projeto serão descritos a seguir:

- ***test_icope_AOLPP_amostras_variaveis_10fold_landmarks15_Loop***: usa marcações faciais para a extração de feições com 15 pontos de marcações. As imagens são organizadas na metodologia de teste 10-fold, divididas em 10 grupos, onde cada grupo é designado para teste e os restantes para treinamento, revezando o grupo de teste. Os experimentos são repetidos 10 vezes. O código fonte deste arquivo está disponível no apêndice A, no final deste trabalho.
- ***test_icope_AOLPP_amostras_variaveis_10fold_imagens_Loop***: usa imagens inteiras para a extração de feições. As imagens são organizadas na metodologia de teste 10-fold, divididas em 10 grupos, onde cada grupo é designado para teste e os restantes para treinamento, revezando o grupo de teste, este teste utiliza imagens inteiras. Os experimentos são repetidos 10 vezes. O código fonte deste arquivo está disponível no apêndice B, no final deste trabalho.

As imagens utilizadas nestes arquivos estão citadas na Figura 5 - “Topologia adotada nas marcações”.

3.1.1 Ambiente de teste (configuração da máquina)

Os testes foram realizados pelo laptop do autor numa configuração que possui um processador i5 de oitava geração modelo 8250u com litografia de 14 nanômetros, com 4 núcleos e 8 Threads, 16GB de RAM utilizando um módulo de memória, SSD de 500GB para armazenamento do Matlab® e dos arquivos necessários para execução do teste e sistema operacional Windows 10.

Cada teste, levou em média de 4 a 6 horas para ser executado, dependendo de quantas imagens ele carregava e realizava os procedimentos necessários para o cálculo.

3.1.2 Quantidade de teste em cada arquivo

Cada arquivo de teste varia dois parâmetros, o PCARatio e o ReducedDim, portanto a quantidade de testes total realizada em cada arquivo de teste é a multiplicação de quantos valores diferentes para o PCARatio vezes a quantidade de valores para o parâmetro ReducedDim. No caso dos testes realizados, nos dois arquivos um de imagens inteiras e outro de marcações foram usados 9 valores diferentes para o PCARatio e 6 valores diferentes para o ReducedDim, portanto, foram realizados 54 testes para cada arquivo de teste.

3.2 Comparação de classificadores SVM e NN

- **Teste com marcações**

- o NN: A taxa máxima de acerto nos testes com esse arquivo usando o classificador do vizinho mais próximo foi 88,57% onde os principais parâmetros foram:

Total de 10 iterações, esquema de cores no espectro RGB foi [1,2,3], a extração de imagens foi definida como “true”, redimensionando assim seu tamanho para [64,48] pixels, no método AOLPP, o PCARATIO foi definido em 0,9, redução de dimensionalidade foi definida em 7, a variável “K” está definida como 54, e o “supervised” definido como verdadeiro (true).

- o SVM: A taxa máxima de acerto nos testes com esse arquivo de teste usando o classificador do SVM foi de 89,52% onde os principais parâmetros foram:

Neste teste, onde foi alcançada a maior taxa até o presente momento, as definições das variáveis foram as seguintes: total de 10 iterações também, espectro RGB [1,2,3]. Na redução de dimensionalidade, os valores foram definidos como PCARATIO em 0,95, variável “K” foi reduzida para 52, o “supervised” neste caso definido como falso (false).

- **Teste com imagens inteiras**

Foram realizadas 10 iterações, com espectro de cores RGB [1,2,3], na seleção de feições o *ResizeImage* foi definido como falso, na Redução de Dimensionalidade, o PCARatio definido em 0,9 a variável *ReducedDim* definida com valor 7, variável “K” = 400, supervised definido como verdadeiro, as técnicas utilizadas foram: 1 (PCA), 2 (LDA) e 7 (AOLPP), para efeitos de comparação das taxas.

Como foi atingido a maior taxa com estes parâmetros, foram mantidos os mesmos para os demais testes, alterando apenas seu tipo e o método utilizado (LDA, PCA, AOLPP).

Os testes utilizando o classificador do vizinho mais próximo (NN) e o método AOLPP obtiveram resultado consideravelmente melhor em comparação às outras técnicas propostas. No classificador NN, a maior taxa obtida foi de 90% e no classificador SVM a maior taxa foi de 89,52%.

Os parâmetros para obtenção destes resultados são os mesmos descritos no primeiro parágrafo deste tópico, assim com estas taxas optou-se por manter já que não foi possível chegar a 100% na taxa de acerto. Estes resultados estão presentes no gráfico da Figura 6 no item 3.6 Gráficos.

3.3 Comparação das técnicas de redução de dimensionalidade (LDA, PCA, AOLPP)

- ***Teste com marcações***

Em todos os arquivos de testes, foram utilizados as 3 técnicas para comparação dos resultados e 2 classificadores (NN e SVM);

Neste arquivo, contém as marcações faciais dos pontos, onde, as variações que apresentaram melhores taxas foram usando o método AOLPP e classificador NN, com os seguintes parâmetros: 10 iterações, espectro de cores RGB [1,2,3], PCARatio 0,9, ReducedDIM 7, variável “K” = 400, e supervised foi definido como verdadeiro (true).

Assim, os únicos valores que foram alterados para efeitos de comparação foram das técnicas 1 (PCA), 2 (LDA e 7 (AOLPP) e os classificadores 1 (NN) e 4 (SVM).

Portanto, nas imagens com marcações alcançou-se uma taxa de 90% no classificador NN e no método AOLPP, tendo sido o melhor resultado alcançado nos testes, já no método LDA a taxa foi de 82,38% e no PCA 84,29%.

Com o classificador SVM as taxas foram no método AOLPP 89,52%, no método PCA 82,38% e no método LDA 86,19%. Então, podemos observar que o melhor caso é usar o classificador NN e o método AOLPP, o segundo melhor caso, é usando o método LDA e o pior cenário seria com o método PCA.

- ***Teste com imagens inteiras***

Neste arquivo de teste, é considerado imagens inteiras para repetir as 10 iterações usando os mesmos parâmetros do arquivo anterior.

Aqui, usando as imagens inteiras para o teste, o método AOLPP também se mostrou mais eficaz em relação às técnicas PCA e LDA. Este método obteve uma taxa de 88,57% também usando o classificador NN, já com o classificador SVM teve uma taxa de 80,95%.

No método LDA com o classificador NN a taxa obtida foi de 85,71% e no classificador SVM, a taxa obtida foi de 88,57%. E no método PCA utilizando o classificador NN a taxa obtida foi de 82,38% e no classificador SVM deu a menor taxa de 80,95%

Então, podemos concluir que utilizando o arquivo de imagens inteiras, o método AOLPP conseguiu uma taxa maior no classificador NN que foi de 90%.

3.4 Influência dos valores dos parâmetros dos testes

- ***Teste com marcações***

Aqui o valor máximo obtido nos testes foi uma taxa de 90% no método AOLPP, com classificador NN.

- PCARatio: os testes indicaram que é necessário um valor alto para esse parâmetro, geralmente em torno dos 0,9(indicando alta preservação do número de dimensões no passo do método PCA). Valores maiores ou menores que este tem apresentado uma tendência de uma taxa de acerto menor. As dimensões descartadas geralmente representam ruído nas imagens.

- Número de Dimensões (reducedDIM) - número de elementos do vetor final, que representa cada imagem, não é necessário um número alto de dimensões, aqui foram consideradas 7 dimensões já que as últimas dimensões, representam elementos não discriminativos das imagens.

- k - quantidade de vizinhos que o método usa para modelar a geometria das classes. Quanto maior o número de vizinhos para modelar a geometria dos dados em uma classe, melhores os resultados. Aqui foram considerados a variável $K=400$.

- ***Teste com imagens inteiras***

Utilizando-se dos mesmos parâmetros do arquivo anterior, os valores que foram definidos nas variáveis PCARatio, ReducedDim e k, neste teste foram obtidos também

através de várias baterias de teste, no Matlab®, onde na maior taxa alcançada, permaneceram-se os valores que foram colocados nas variáveis finais.

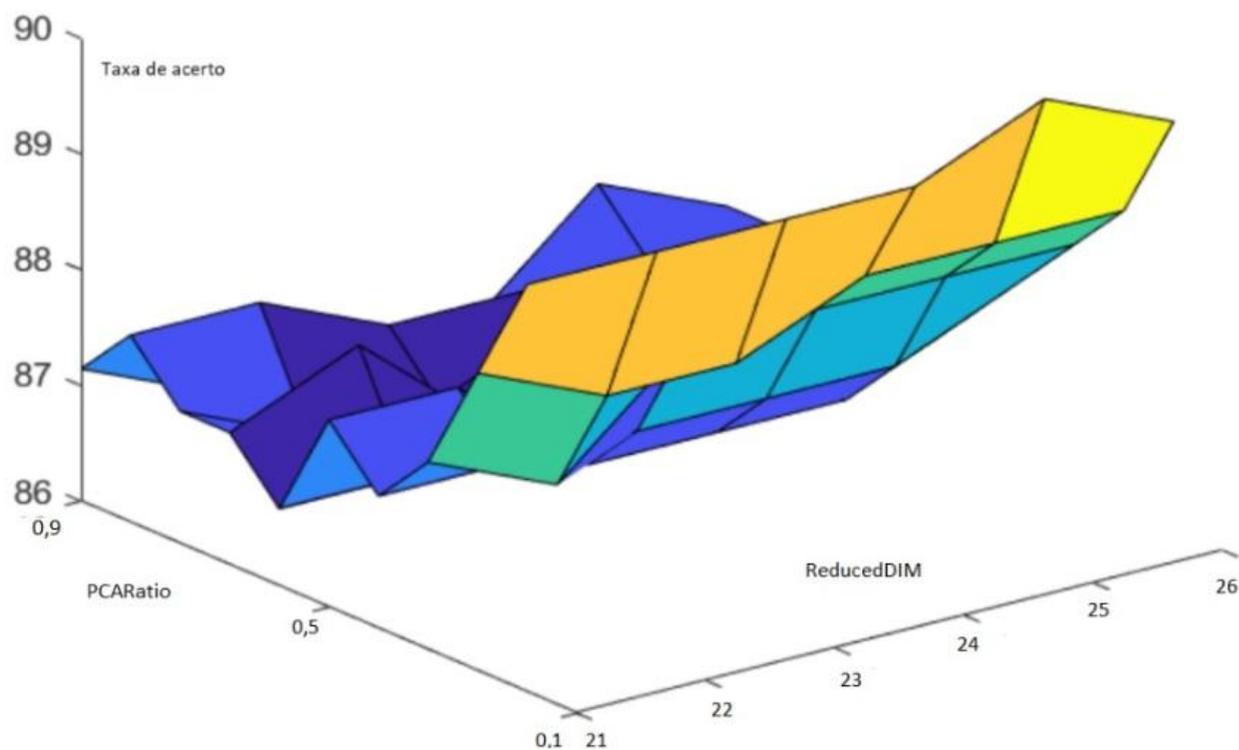
Neste arquivo, também se obteve melhor resultado utilizando o método AOLPP e o classificador NN, também se manteve os valores das variáveis PCARatio, ReducedDIM e a variável k igual do arquivo anterior.

3.6 Gráficos

A seguir, serão mostrados gráficos que foram obtidos através de testes rodados nos arquivos de imagens inteiras, e imagens com marcações, o primeiro contém imagens inteiras para executar os testes, e o segundo arquivo utiliza os 15 pontos marcados nas imagens para realização dos testes. Cada teste, destes 2 arquivos foram repetidos 10 vezes em suas execuções.

Nas figuras 6 e 7, houve uma configuração de parâmetros que levou ao maior valor da taxa de acerto (eixo z), a qual foi apresentada anteriormente neste trabalho, com eixos x e y indicando variações do PCARatio e reducedDim.

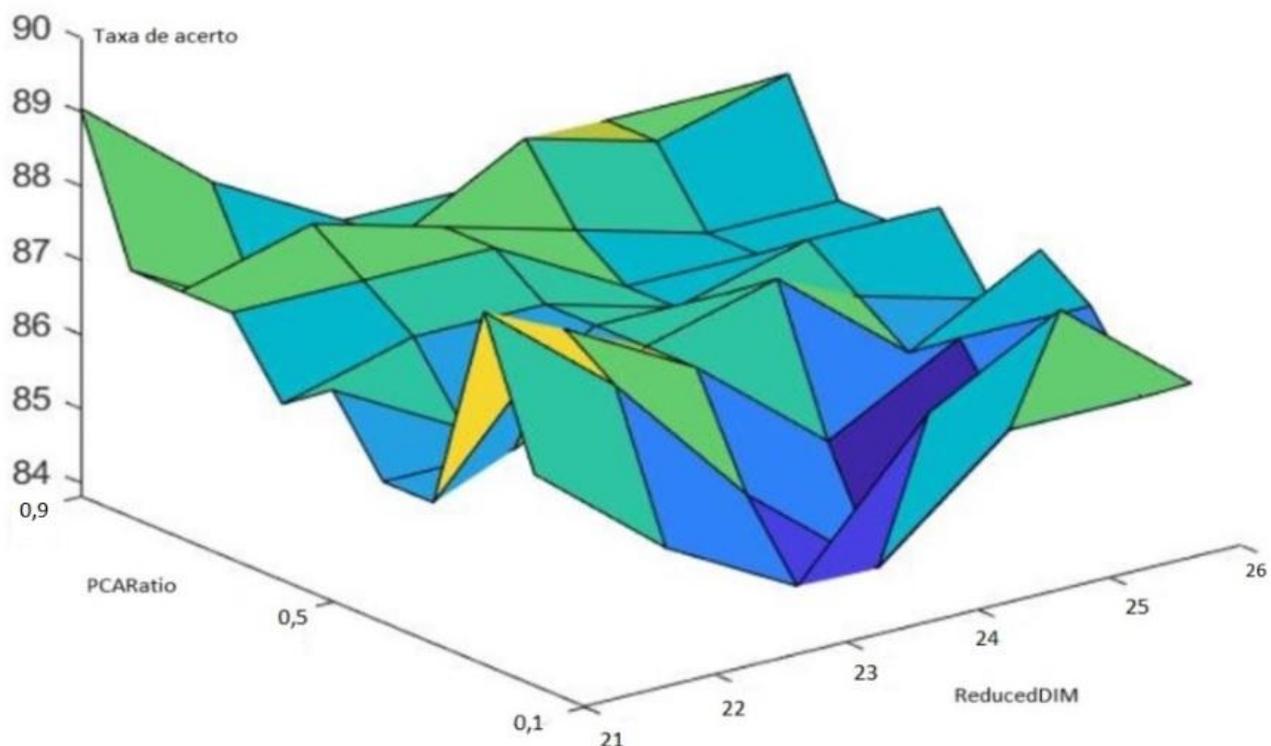
Figura 6: Gráfico de um conjunto de resultados obtidos com a execução do arquivo de imagens inteiras



Fonte: Elaborado pelo Autor, 2022

Neste gráfico da figura 6, temos os resultados atingidos nos testes do arquivo de imagens inteiras, seguindo a linha dos eixos x PCARatio e y ReducedDIM, com o eixo z na vertical sendo as taxas de acerto.

Figura 7: Gráfico de um conjunto de resultados obtidos com a execução do arquivo com marcações



Fonte: Elaborado pelo Autor, 2022

Na figura 7, é apresentado o gráfico do arquivo de marcações faciais, utilizando as imagens com os 15 pontos marcados.

Estes gráficos são conhecidos como uma malha de coordenadas em intervalos específicos. As cores são geradas automaticamente pelo Matlab® a fim de que o gráfico seja melhor visualizado.

4 CONSIDERAÇÕES FINAIS

Estes resultados vistos anteriormente serviram de apoio para o projeto de reconhecimento facial e os dados obtidos foram incluídos em uma base no Matlab®, estas taxas de acerto vistas na seção de resultados e discussões, possibilitaram a continuação do projeto de pesquisa já que se obteve um resultado satisfatório neste ramo da pesquisa.

Para que fossem encontrados estes valores em porcentagem, foram necessárias várias baterias de testes no Matlab®, realizando possíveis ajustes no código, alterando algumas variáveis e executando os testes novamente para assim comprovar se houve aumento ou diminuição das porcentagens.

Caso o resultado destas alterações fosse positivo (aumento das porcentagens), era mantido o valor das variáveis que forneceram este resultado, caso contrário, foram realizadas outras baterias de testes com mais mudança de valores até que os resultados finais correspondessem com o esperado nos testes.

Assim, com apoio das planilhas de anotações dos resultados e dos gráficos elaborados pelo Matlab®, obteve-se uma melhora considerável dos resultados desde o primeiro teste realizado, até os últimos destes dois arquivos que serviram de base para este Trabalho de Conclusão de Curso. Estes arquivos serviram de base para comparação de diversas técnicas utilizadas entre eles: o PCA, LDA e AOLPP, o qual o terceiro apresentou melhor resultado em comparação aos outros dois, utilizando o classificador de vizinho mais próximo (NN).

Verificou-se que não haviam muitos trabalhos relacionados ao tema de reconhecimento de dor, assim surgiu-se a ideia deste trabalho para além de maximizar as taxas de acerto auxiliar outras pessoas que se interessam por este tema, pois foi verificado que estas taxas de acerto deste trabalho de conclusão de curso foram satisfatórias no âmbito de aplicar técnicas de reconhecimento de dor.

No apêndice C, é apresentado um artigo desenvolvido por (SOLDERA, J., 2015) denominado “Reconhecimento de dor em recém-nascidos”, para a Conferência Brasileira de Informática em Saúde CBIS no qual apresenta resultados parciais atingidos durante a execução deste projeto (SOLDERA, J. et. al. 2021).

REFERÊNCIAS

BELHUMEUR, P. N.; HESPANHA, J. a. P; KRIEGMAN, D.J. Eigenfaces vs Fisherfaces: recognition using class specific linear projection. *IEEE transactions on Pattern Analysis and Machine Intelligence*, [S.1.] V.19, N.7, P.711-720, Julho 1997.

BUZUTI, L.; HEIDEIRICH, T.; BARROS, M.; GUINSBURG, R.; THOMAZ, C.; Neonatal Pain Assessment From Facial Expression Using Deep Neural Networks. *In: WORKSHOP DE VISÃO COMPUTACIONAL (WVC)*, 16. , 2020, Evento Online. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2020, p. 87-92. Acesso em 13 jul. 2022.

BUZUTI, L, THOMAS, C. E. Avaliação de Dor em Expressão Facial Neonatal por meio de Redes Neurais Profundas, Departamento de Engenharia Elétrica – Laboratório de Processamento de Imagens Centro Universitário FEI – São Bernardo do Campo – SP – Brasil, 2020.

BRAHNAM, Sheryl et al. Machine recognition and representation of neonatal facial displays of acute pain. *Artificial intelligence in medicine*, v. 36, n. 3, p. 211-222, 2006.

BRANHAM, S., Chuang, C.-F., Shih, F. Y., and Slack, M. R. (2006). Machine recognition and representation of neonatal facial displays of acute pain. *Artificial intelligence in medicine*, 36(3):211–222.

CAI, D.; X.; HAN, J.; ZHANG, H.-J, Orthogonal laplacianfaces for face recognition. *IEEE Transactions on Image Processing*, [S.I.] v.15, n.11, p.3608-3614, Novembro, 2006.

COVER, T.; HART, P. Nearest neighbor pattern classification, *IEEE Transactions on Information Theory*. [S.I], v.13, n.1, p.21-27, Janeiro, 1967.

DARWIN, C. A Origem das Espécies. Hemus – Livraria Editora Ltda, 1859, São Paulo, SP.

FIE-FIE, L.; LI, K. ImageNet. Disponível em: <<https://image-net.org/>>, 2016. Acesso em: 27 de set. 2022.

GALVAN, Carina et al. A efetividade do tratamento da dor no pós operatório de cirurgias ortopédicas. **Revista Eletrônica Acervo Saúde**, v. 12, n. 10, p. e4875-e4875, 2020.

G. Zamzmi, R. Paul, D. Goldgof, R. Kasturi, and Y. Sun, "Pain assessment from facial expression: Neonatal convolutional neural network (n-cnn)," in 2019 International Joint Conference on Neural Networks (IJCNN). IEEE, 2019, pp. 1–7.

HE, X.; YAN. S.; HU, Y.; NIYOGI, P.; ZHANG, H.jiang. Face recognition using laplacianfaces. IEEE transactions on Pattern Analysis and Machine Intelligence, [S.1], V.27, N.3, P.328-340, Março, 2005.

INTERNATIONAL ASSOCIATION FOR THE STUDY OF PAIN SUBCOMMITTEE ON TAXONOMY-PAIN TERMS: A list with definitions and notes on usage. Pain, 6:249 – 52, 1979.

MATLAB®, 2022, Disponível em: <<https://la.mathworks.com/products/matlab.html>> , Acesso em 27 jun. 2021.

Resnet 50 Rede Neural Convolutacional, 2022, Disponível em: <<https://www.mathworks.com/help/deeplearning/ref/resnet50.html;jsessionid=03d4e0681150d2523a11e84b5e35>> Acesso em: 18 set. 2022.

RCX Balda, R Guinsburg, A linguagem da dor no recém-nascido, 2018. Disponível em: <https://www.sbp.com.br/fileadmin/user_upload/DocCient-Neonatal-Linguagem_da_Dor_atualizDEz18.pdf> Acesso em: 06 out. 2021.

SBIS, Sociedade Brasileira de Informática em Saúde, Disponível em: <<http://sbis.org.br/cbis-2022/>>, Acesso em 23 set. 2022

SOLDERA, J., Reconhecimento Facial com Projeções Ortogonais Preservadoras de Localidade Customizadas para Maximizar Imagens Suaves, Tese de Doutorado, Porto Alegre, 2015.

SOLDERA, J., C. Alberto Ramirez Behaine and J. Scharcanski, "Customized Orthogonal Locality Preserving Projections With Soft-Margin Maximization for Face Recognition," in IEEE Transactions on Instrumentation and Measurement, vol. 64, no. 9, pp. 2417-2426, setembro, 2015.

SOLDERA, J., J. F. A. Lucca, Becker, M. L., Kreuzberg, R. Rothe, W. F., Comaru, T., Soldera, J. Reconhecimento Automático de dor em recém-nascidos, Conferência Brasileira de Informática em Saúde, CBIS, 2021.

PACHECO, A. K vizinhos mais próximos - KNN, Disponível em: <<http://computacaointeligente.com.br/algoritmos/k-vizinhos-mais-proximos/>>, Acesso em 24 jun. 2022.

PENHARBEL, Eder Augusto et al. E-Faces-Um classificador capaz de analisar imagens e classificá-las como faces ou não faces utilizando o método Eigenfaces. I Workshop de Visão Computacional, p. 13, 2005.

T. M. Heiderich, A. T. F. S. Leslie, and R. Guinsburg, “Neonatal procedural pain can be assessed by computer software that has good sensitivity and specificity to detect facial movements,” *Acta Paediatrica*, vol. 104, no. 2, pp. e63–e69, 2015.

VAPNIK, V. N. *Statistical Learning Theory*. New York: Wiley-Interscience, 1998.

Viola, P., Jones, M.J. Robust Real-Time Face Detection. *International Journal of Computer Vision* 57, 137–154, 2004.

APÊNDICE A

CÓDIGO DA PROGRAMAÇÃO UTILIZADO NO ARQUIVO DE PONTOS DE MARCAÇÕES FACIAIS

```
%% Configurações Iniciais do Teste:
```

```
% Limpa as variáveis de configuração do teste:
```

```
clear parameters;
```

```
% Imprime esse arquivo no console (para exibição dos parâmetros do teste):
```

```
type ([mfilename('fullpath'), '.m'])
```

```
% Carrega configurações de diretório globais:
```

```
loadGlobalVariables;
```

```
param1 = [0.96:0.01:1];
```

```
param2 = [6:1:20];
```

```
%% Configurações da base de imagens usada no teste:
```

```
% Identificador da base: 12: facesdb
```

```
database = 13;
```

```
% Modalidade de carregamento das imagens da base (1: padrão).
```

```
modality = 3;
```

```
% Carrega as configurações da base de imagens:
```

```
databaseDescriptor = loadDatabaseDescriptor(database, modality);
```

```
% Especifica quais são as classes de imagens existentes na base (expressões faciais)
```

```
parameters.classes = {'_nonpain', '_pain'};
```

```
% Especifica quantas imagens por classe são disponíveis

%parameters.images = num2cell(1:204);% all

parameters.images{1} = num2cell(1:144);% nonpain

parameters.images{2} = num2cell(1:60);%pain

parameters.imagesTotal = 144+60;

%% Define parâmetros de organização do teste:

% Número de imagens reservadas para treinamento das 36 (por classe):

parameters.trainingImages = 0;

% Número de imagens reservadas para teste das 36 (por classe):

parameters.testImages = 0;

% Modo de seleção de imagens:

% 0: sempre seleciona imagens aleatórias a cada repetição do teste.

% 1: seleciona imagens aleatórias no primeiro teste e mantém a mesma seleção de imagens
nos próximos testes

parameters.imageSelection.mode = 0;

% Tipo de validação de teste:

% false: não realiza validação cruzada (modo normal de teste)

% true: usa o método de validação cruzada (o número de imagens de teste e treinamento
deverão ser ambos zero)

% parameters.trainingImages=0 e parameters.testImages=0
```

```
parameters.crossValidation = true;

% número de repetições se usada validação cruzada (k-fold):

parameters.kFold = 10;

% Se auto carrega a mesma seleção de imagens aleatória que o teste anterior

parameters.imageSelection.autoload = true;

% Se auto salva a seleção de imagens aleatórias usada neste teste

parameters.imageSelection.autosave = true;

%% Se repete o teste definido novamente

% Número de iterações do mesmo teste

parameters.iterations = 20;

% define as interações específicas para serem executadas (todas: em branco)

parameters.iterationSet = []; % Specific iterations.

%% Parâmetros de seleção de feições (características) das imagens:

% Canais de cores analisados:

% 1,2,3 = RGB (1: red, 2: green, 3: blue).

% 4,5,6 = Gradientes do RGB (variação de cor).

% 0 = escalas de cinza.

% 7,8,9 = Espaço de cores "Lab"

% 10,11,12: Espaço de cores: "normalized RGB"

% 13: gradiente da imagem em escalas de cinza (primeira ordem)
```

```

% 14: gradiente da imagem em escalas de cinza (segunda ordem)

% 15: gradiente da imagem em escalas de cinza (terceira ordem)

% 20,21,22: Gradientes dos Canais de Cores RGB

% 23,24,25: Derivada da imagem em RGB na direção do eixo X

% 26,27,28: Derivada da imagem em RGB na direção do eixo Y

% 30,31,32: Imagem Integral RGB na direção do eixo X

% 33,34,35: Imagem Integral RGB na direção do eixo Y

% 40,41,42: Representação em LogPolar (RGB)

% 43,44,45: Local Binary Pattern - LBP (Método 1).

% 46,47,48: Local Binary Pattern - LBP (Método 2).

% [61,62,63]: Cosine Transform

% [64,65,66]: Log Cosine Transform

% [67,68,69,70]: single_scale_self_quotient_image

% 101, 102, 103, ...: multiscale image representation (101: first level, 102: second level, ...)

% 211, 212, 213: 2XY, neighborhood variance image: where X=radius, Y=channel (1:red,
2:green, 3:blue)

parameters.featureExtraction.raw.colorChannels = [79,80,81,83,84,85];
%[75,76,77];%[1,2,3];%[43,44,45]; %[20,21,22]; %[1,2,3]=RGB

% Se força a conversão para um formato numérico específico

% true: sim

% false: não

```

```
parameters.featureExtraction.raw.forceNumberFormat = false;

% Qual formato numérico que ocorre a conversão:

parameters.featureExtraction.raw.numberFormat = 'uint8';

% se usa pesos nas feições (multiplicadores):

parameters.featureExtraction.raw.adjustWeights = false;

% pesos dos canais de cor: (1: sem mudança):

parameters.featureExtraction.raw.weights    = [1 1 1];

% se redimensiona imagens:

% true: muda o tamanho

% false: mantém o tamanho original das imagens

parameters.featureExtraction.raw.resizeImage = false;

% se rotaciona imagens que estão em paisagem.

parameters.featureExtraction.raw.rotate = false;

% novo tamanho de imagem se selecionado parâmetro para redimensionar:

parameters.featureExtraction.raw.newImageSize = [64, 48];

% tipo de normalização das feições da imagem:

% 0: sem normalização

% 1: normalização (torna média 0 e desvio padrão 1):

parameters.featureExtraction.raw.normalizationType = 0;

% autoload da extração de feições:
```

```

parameters.featureExtraction.raw.autoload = true;

% autosave da extração de feições:

parameters.featureExtraction.raw.autosave = true;

%% Extração de Landmarks:

parameters.featureExtraction.processed.landmarks.desenhar = false;

parameters.featureExtraction.processed.landmarks.landmarks = [1:8];

parameters.featureExtraction.processed.landmarks.interpolate = true;

parameters.featureExtraction.processed.landmarks.interpolatedLandmarkNumber = 5;

parameters.featureExtraction.processed.landmarks.interpolationMode = 5;

parameters.featureExtraction.processed.landmarks.tamanhoJanela = 10;

parameters.featureExtraction.processed.landmarks.modosNormalizar = 0;

%% Opções avançadas de pós-processamento das feições extraídas anteriormente:

% feature extraction options

parameters.featureExtraction.processed.method = 3; %1 = vectorized image concatenating
pixels from all color channels, 2=rafael, 3=gaussians %4=8 gaussians %5=block image
%6=foveation %7: texture feature extraction plus random projections (use block parameters)

parameters.featureExtraction.processed.forceNumberFormat = false;

parameters.featureExtraction.processed.numberFormat = 'uint8';

parameters.featureExtraction.processed.concatenateColorChannels = false;

parameters.featureExtraction.processed.normalization=0; %0: no normalization, 1: weber's
law, 2: euclidean

```

parameters.featureExtraction.processed.imageBlocks.blockSelectionMode = 1; % 0: Uses block size and pace to get the blocks. 1: random blocks (numberOfBlocks)

parameters.featureExtraction.processed.imageBlocks.numberOfBlocks = 3000;

parameters.featureExtraction.processed.imageBlocks.size = [3 3];

parameters.featureExtraction.processed.imageBlocks.pace = 4;

parameters.featureExtraction.processed.imageBlocks.modes = [3]; %1:% Sorted Pixels Values: (Global) %2: Sorted Pixels Values: (Square) %3: Sorted Pixel Values: Circular %4: Sorted Pixel Differences: Radial-diff %5: Sorted Pixel Differences: Angular-diff

parameters.featureExtraction.processed.imageBlocks.normalizationStep=0; %0: normalize color channels independently, 1: normalize the whole random feature vector

parameters.featureExtraction.processed.imageBlocks.sortPixelMode = 1; % sort mode from pixels extracted from each block. 0: no sort, 1: sort each circle

parameters.featureExtraction.processed.imageBlocks.textons.apply = 0; % if uses textons (1: true)

parameters.featureExtraction.processed.imageBlocks.textons.AOLPP.apply = false; % if it compacts the histograms

parameters.featureExtraction.processed.imageBlocks.textons.AOLPP.options.PCARatio = 0.9;

parameters.featureExtraction.processed.imageBlocks.textons.AOLPP.options.ReducedDim = 5;

parameters.featureExtraction.processed.imageBlocks.textons.AOLPP.options.Metric = 'Euclidean';

parameters.featureExtraction.processed.imageBlocks.textons.AOLPP.localityScheme = 1;

parameters.featureExtraction.processed.imageBlocks.textons.AOLPP.supervised = true;

```

parameters.featureExtraction.processed.imageBlocks.textons.reduceTextons = true;

parameters.featureExtraction.processed.imageBlocks.textons.textonDivisor = 4;

parameters.featureExtraction.processed.imageBlocks.textons.kTextons = 100;

parameters.featureExtraction.processed.imageBlocks.textons.clusterizationMethod =
2;%1:k-means 2: kmeans lite, 3: matlab k-means, 4: LSC, 5:Hiarquical clustering

parameters.featureExtraction.processed.imageBlocks.textons.graphClusterMetric =
1;%1:euclidian, 2: cosine

parameters.featureExtraction.processed.imageBlocks.textons.landmarkNumber = 1000;

parameters.featureExtraction.processed.imageBlocks.textons.maxIter = 50;

parameters.featureExtraction.processed.randomProjection.apply = false;

parameters.featureExtraction.processed.randomProjection.dims = 10;

parameters.featureExtraction.processed.randomProjection.percentual = false;%
percentual=true kept dimensions: between 0 and 1, false: dimension value

parameters.featureExtraction.processed.randomProjection.fixedMatrix = true;

parameters.featureExtraction.processed.randomProjection.separate = true; % creates diferent
modes for each high-dimensional dimension

parameters.featureExtraction.processed.randomProjection.mode = 1; %0: 0 and 1 only.
1:sqrt(3)*{0,-1,1} 2:{-1,+1}

parameters.featureExtraction.processed.randomProjection.sortBeforeRP = false; %0: don't
sort pixel values (just concatenate), 1: sort in order to concatenate

parameters.featureExtraction.processed.randomProjection.sortAfterRP = false; % if sorts the
resulting feature vector

parameters.featureExtraction.processed.autoload = true;

```

```
parameters.featureExtraction.processed.autosave = true;

% dimensionality reduction options

parameters.dimensionalityReduction.method = 1; % 0=none, 1=PCA, 2=LDA %7=AOLPP

parameters.dimensionalityReduction.cuda = false;

%parameters.dimensionalityReduction.PCA.options.ReducedDim = 20;

parameters.dimensionalityReduction.PCA.options.PCARatio = 0.9;

parameters.dimensionalityReduction.LDA.options.PCARatio = 0.9;

parameters.dimensionalityReduction.LDA.options.Fisherface = 1;

parameters.dimensionalityReduction.AOLPP.options.PCARatio = 1;

parameters.dimensionalityReduction.AOLPP.options.ReducedDim = 7;

parameters.dimensionalityReduction.AOLPP.options.k = 54;

parameters.dimensionalityReduction.AOLPP.localityScheme = 4; %1: Heat Kernel, 2: Induct
LDA 3: geodesics % 4: OLPP LDA (1/ni)

parameters.dimensionalityReduction.AOLPP.supervised = true;

parameters.dimensionalityReduction.autoload = false;

parameters.dimensionalityReduction.autosave = false;

% Classification method:

parameters.classification.method = 4; %1: NN (nearest neighbor), 2: inverse NN, 3:KNN, 4:
SVM soft margins 5: ERM (Ranking) %100=single Dodson geodesics % 101:KL

parameters.classification.embeddSVM = false;

parameters.classification.SVM.soft = true;
```

```

parameters.classification.SVM.kernelMode = 1; %1: linear SVM (linear product kernel)

parameters.classification.SVM.sigma = 0.5; %1: RBF kernel parameter (DEVIATION) %para
RGB: 0.000068 ou 69

parameters.classification.SVM.C = 1;

parameters.classification.SVM.supervised = true;

parameters.classification.SVM.histograms.globalLimit = false;

parameters.classification.SVM.histograms.bins = 50;

parameters.classification.EMR.landmarkNumber = 150;

parameters.classification.histograms.bins = 50;

parameters.classification.histograms.globalLimit = false;

% global autoload/autosave

parameters.autoload = false;

parameters.autosave = true;

% Run a method using the selected database

% Variable: variable number of samples per class

result = zeros(length(param1), length(param2));

for count1 = 1 : length(param1)

    for count2 = 1 : length(param2)

        parameters.dimensionalityReduction.PCA.options.PCARatio = param1(count1);

        parameters.dimensionalityReduction.PCA.options.ReducedDim = param2(count2);

        result(count1, count2) = runApplicationVariable(databaseDescriptor, parameters);

```

```
end
```

```
end
```

```
max(result(:))
```

```
disp('fim');
```

APÊNDICE B

CÓDIGO FONTE DO ARQUIVO DE IMAGENS INTEIRAS

```
%% Configurações Iniciais do Teste:

% Limpa as variáveis de configuração do teste:

clear parameters;

% Imprime esse arquivo no console (para exibição dos parâmetros do teste):

type ([mfilename('fullpath'), '.m'])

% Carrega configurações de diretório globais:

loadGlobalVariables;

param1 = [0.96:0.01:1];

param2 = [6:1:20];

%% Configurações da base de imagens usada no teste:

% Identificador da base: 12: facesdb

database = 13;

% Modalidade de carregamento das imagens da base (1: padrão).

modality = 3;

% Carrega as configurações da base de imagens:

databaseDescriptor = loadDatabaseDescriptor(database, modality);

% Especifica quais são as classes de imagens existentes na base (expressões faciais)

parameters.classes = {'_nonpain', '_pain'};
```

```
%Especifica quantas imagens por classe são disponíveis

%parameters.images = num2cell(1:204);% all

parameters.images{1} = num2cell(1:144);% nonpain

parameters.images{2} = num2cell(1:60);%pain

parameters.imagesTotal = 144+60;

%% Define parâmetros de organização do teste:

% Número de imagens reservadas para treinamento das 36 (por classe):

parameters.trainingImages = 0;

% Número de imagens reservadas para teste das 36 (por classe):

parameters.testImages = 0;

% Modo de seleção de imagens:

% 0: sempre seleciona imagens aleatórias a cada repetição do teste.

% 1: seleciona imagens aleatórias no primeiro teste e mantém a mesma seleção de imagens
nos próximos testes

parameters.imageSelection.mode = 0;

% Tipo de validação de teste:

% false: não realiza validação cruzada (modo normal de teste)

% true: usa o método de validação cruzada (o número de imagens de teste e treinamento
deverão ser ambos zero)

% parameters.trainingImages=0 e parameters.testImages=0
```

```
parameters.crossValidation = true;

% número de repetições se usada validação ocruzada (k-fold):

parameters.kFold = 10;

% Se autocarrega a mesma seleção de imagens aleatória que o teste anterior

parameters.imageSelection.autoload = true;

% Se autosalva a seleção de imagens aleatórias usada nesse teste

parameters.imageSelection.autosave = true;

%% Se repete o teste definido novamente

% Número de iterações do mesmo teste

parameters.iterations = 20;

% define as iterações específicas para serem executadas (todas: em branco)

parameters.iterationSet = []; % Specific iterations.

%% Parâmetros de seleção de feições (características) das imagens:

% Canais de cores analisados:

% 1,2,3 = RGB (1: red, 2: green, 3: blue).

% 4,5,6 = Gradientes do RGB (variação de cor).

% 0 = escalas de cinza.

% 7,8,9 = Espaço de cores "Lab"

% 10,11,12: Espaço de cores: "normalized RGB"

% 13: gradiente da imagem em escalas de cinza (primeira ordem)
```

```

% 14: gradiente da imagem em escalas de cinza (segunda ordem)

% 15: gradiente da imagem em escalas de cinza (terceira ordem)

% 20,21,22: Gradientes dos Canais de Cores RGB

% 23,24,25: Derivada da imagem em RGB na direção do eixo X

% 26,27,28: Derivada da imagem em RGB na direção do eixo Y

% 30,31,32: Imagem Integral RGB na direção do eixo X

% 33,34,35: Imagem Integral RGB na direção do eixo Y

% 40,41,42: Representação em LogPolar (RGB)

% 43,44,45: Local Binary Pattern - LBP (Método 1).

% 46,47,48: Local Binary Pattern - LBP (Método 2).

% [61,62,63]: Cosine Transform

% [64,65,66]: Log Cosine Transform

% [67,68,69,70]: single_scale_self_quotient_image

% 101, 102, 103, ...: multiscale image representation (101: first level, 102: second level, ...)

% 211, 212, 213: 2XY, neighborhood variance image: where X=radius, Y=channel (1:red,
2:green, 3:blue)

parameters.featureExtraction.raw.colorChannels = [79,80,81,83,84,85];
%[75,76,77];%[1,2,3];%[43,44,45]; %[20,21,22]; %[1,2,3]=RGB

% Se força a conversão para um formato numérico específico

% true: sim

% false: não

```

```
parameters.featureExtraction.raw.forceNumberFormat = false;

% Qual formato numérico que ocorre a conversão:

parameters.featureExtraction.raw.numberFormat = 'uint8';

% se usa pesos nas feições (multiplicadores):

parameters.featureExtraction.raw.adjustWeights = false;

% pesos dos canais de cor: (1: sem mudança):

parameters.featureExtraction.raw.weights    = [1 1 1];

% se redimensiona imagens:

% true: muda o tamanho

% false: mantém o tamanho original das imagens

parameters.featureExtraction.raw.resizeImage = false;

% se rotaciona imagens que estão em paisagem.

parameters.featureExtraction.raw.rotate = false;

% novo tamanho de imagem se selecionado parâmetro para redimensionar:

parameters.featureExtraction.raw.newImageSize = [64, 48];

% tipo de normalização das feições da imagem:

% 0: sem normalização

% 1: normalização (torna média 0 e desvio padrão 1):

parameters.featureExtraction.raw.normalizationType = 0;

% autoload da extração de feições:
```

```
parameters.featureExtraction.raw.autoload = true;

% autosave da extração de feições:

parameters.featureExtraction.raw.autosave = true;

%% Extração de Landmarks:

parameters.featureExtraction.processed.landmarks.desenhar = false;

parameters.featureExtraction.processed.landmarks.landmarks = [1:8];

parameters.featureExtraction.processed.landmarks.interpolate = true;

parameters.featureExtraction.processed.landmarks.interpolatedLandmarkNumber = 5;

parameters.featureExtraction.processed.landmarks.interpolationMode = 5;

parameters.featureExtraction.processed.landmarks.tamanhoJanela = 10;

parameters.featureExtraction.processed.landmarks.modosNormalizar = 0;

%% Opções avançadas de pós-processamento das feições extraídas anteriormente:

% feature extraction options

parameters.featureExtraction.processed.method = 3; %1 = vectorized image concatenating
pixels from all color channels, 2=rafael, 3=gaussians %4=8 gaussians %5=block image
%6=foveation %7: texture feature extraction plus random projections (use block parameters)

parameters.featureExtraction.processed.forceNumberFormat = false;

parameters.featureExtraction.processed.numberFormat = 'uint8';

parameters.featureExtraction.processed.concatenateColorChannels = false;

parameters.featureExtraction.processed.normalization=0; %0: no normalization, 1: weber's
law, 2: euclidean
```

parameters.featureExtraction.processed.imageBlocks.blockSelectionMode = 1; % 0: Uses block size and pace to get the blocks. 1: random blocks (numberOfBlocks)

parameters.featureExtraction.processed.imageBlocks.numberOfBlocks = 3000;

parameters.featureExtraction.processed.imageBlocks.size = [3 3];

parameters.featureExtraction.processed.imageBlocks.pace = 4;

parameters.featureExtraction.processed.imageBlocks.modes = [3]; %1:% Sorted Pixels Values: (Global) %2: Sorted Pixels Values: (Square) %3: Sorted Pixel Values: Circular %4: Sorted Pixel Differences: Radial-diff %5: Sorted Pixel Differences: Angular-diff

parameters.featureExtraction.processed.imageBlocks.normalizationStep=0; %0: normalize color channels independently, 1: normalize the whole random feature vector

parameters.featureExtraction.processed.imageBlocks.sortPixelMode = 1; % sort mode from pixels extracted from each block. 0: no sort, 1: sort each circle

parameters.featureExtraction.processed.imageBlocks.textons.apply = 0; % if uses textons (1: true)

parameters.featureExtraction.processed.imageBlocks.textons.AOLPP.apply = false; % if it compacts the histograms

parameters.featureExtraction.processed.imageBlocks.textons.AOLPP.options.PCARatio = 0.9;

parameters.featureExtraction.processed.imageBlocks.textons.AOLPP.options.ReducedDim = 5;

parameters.featureExtraction.processed.imageBlocks.textons.AOLPP.options.Metric = 'Euclidean';

parameters.featureExtraction.processed.imageBlocks.textons.AOLPP.localityScheme = 1;

parameters.featureExtraction.processed.imageBlocks.textons.AOLPP.supervised = true;

```

parameters.featureExtraction.processed.imageBlocks.textons.reduceTextons = true;

parameters.featureExtraction.processed.imageBlocks.textons.textonDivisor = 4;

parameters.featureExtraction.processed.imageBlocks.textons.kTextons = 100;

parameters.featureExtraction.processed.imageBlocks.textons.clusterizationMethod =
2;%1:k-means 2: kmeans lite, 3: matlab k-means, 4: LSC, 5:Hiarquical clustering

parameters.featureExtraction.processed.imageBlocks.textons.graphClusterMetric =
1;%1:euclidian, 2: cosine

parameters.featureExtraction.processed.imageBlocks.textons.landmarkNumber = 1000;

parameters.featureExtraction.processed.imageBlocks.textons.maxIter = 50;

parameters.featureExtraction.processed.randomProjection.apply = false;

parameters.featureExtraction.processed.randomProjection.dims = 10;

parameters.featureExtraction.processed.randomProjection.percentual = false;%
percentual=true kept dimensions: between 0 and 1, false: dimension value

parameters.featureExtraction.processed.randomProjection.fixedMatrix = true;

parameters.featureExtraction.processed.randomProjection.separate = true; % creates diferent
modes for each high-dimensional dimension

parameters.featureExtraction.processed.randomProjection.mode = 1; %0: 0 and 1 only.
1:sqrt(3)*{0,-1,1} 2:{-1,+1}

parameters.featureExtraction.processed.randomProjection.sortBeforeRP = false; %0: don't
sort pixel values (just concatenate), 1: sort in order to concatenate

parameters.featureExtraction.processed.randomProjection.sortAfterRP = false; % if sorts the
resulting feature vector

parameters.featureExtraction.processed.autoload = true;

```

```
parameters.featureExtraction.processed.autosave = true;

% dimensionality reduction options

parameters.dimensionalityReduction.method = 1; % 0=none, 1=PCA, 2=LDA %7=AOLPP

parameters.dimensionalityReduction.cuda = false;

%parameters.dimensionalityReduction.PCA.options.ReducedDim = 20;

parameters.dimensionalityReduction.PCA.options.PCARatio = 0.9;

parameters.dimensionalityReduction.LDA.options.PCARatio = 0.9;

parameters.dimensionalityReduction.LDA.options.Fisherface = 1;

parameters.dimensionalityReduction.AOLPP.options.PCARatio = 1;

parameters.dimensionalityReduction.AOLPP.options.ReducedDim = 7;

parameters.dimensionalityReduction.AOLPP.options.k = 54;

parameters.dimensionalityReduction.AOLPP.localityScheme = 4; %1: Heat Kernel, 2: Induct
LDA 3: geodesics % 4: OLPP LDA (1/ni)

parameters.dimensionalityReduction.AOLPP.supervised = true;

parameters.dimensionalityReduction.autoload = false;

parameters.dimensionalityReduction.autosave = false;

% Classification method:

parameters.classification.method = 4; %1: NN (nearest neighbor), 2: inverse NN, 3:KNN, 4:
SVM soft margins 5: ERM (Ranking) %100=single Dodson geodesics % 101:KL

parameters.classification.embeddSVM = false;

parameters.classification.SVM.soft = true;
```

```

parameters.classification.SVM.kernelMode = 1; %1: linear SVM (linear product kernel)

parameters.classification.SVM.sigma = 0.5; %1: RBF kernel parameter (DEVIATION) %para
RGB: 0.000068 ou 69

parameters.classification.SVM.C = 1;

parameters.classification.SVM.supervised = true;

parameters.classification.SVM.histograms.globalLimit = false;

parameters.classification.SVM.histograms.bins = 50;

parameters.classification.EMR.landmarkNumber = 150;

parameters.classification.histograms.bins = 50;

parameters.classification.histograms.globalLimit = false;

% global autoload/autosave

parameters.autoload = false;

parameters.autosave = true;

% Run a method using the selected database

% Variable: variable number of samples per class

result = zeros(length(param1), length(param2));

for count1 = 1 : length(param1)

    for count2 = 1 : length(param2)

        parameters.dimensionalityReduction.PCA.options.PCARatio = param1(count1);

        parameters.dimensionalityReduction.PCA.options.ReducedDim = param2(count2);

        result(count1, count2) = runApplicationVariable(databaseDescriptor, parameters);

```

```
end
```

```
end
```

```
max(result(:))
```

```
disp('fim');
```

APÊNDICE C

Reconhecimento Automático de Dor em Recém-Nascidos

J. F. A. Lucca¹, Becker, M. L.², Kreuzberg, R.³ Rothe, W. F.⁴, Comaru, T.⁵,
Soldara, J.⁶

¹Instituto Federal Farroupilha, Santo Ângelo, RS

²Instituto Federal Farroupilha, Santo Ângelo, RS

³Instituto Federal Farroupilha, Santo Ângelo, RS

⁴Instituto Federal Farroupilha, Santo Ângelo, RS

⁵Instituto Federal Farroupilha, Santo Ângelo, RS

⁶Instituto Federal Farroupilha, Santo Ângelo, RS

jose.2020305158@aluno.iffar.edu.br, maickel.2019004732@aluno.iffar.edu.br,
kruzbergritieli@gmail.com, willianrohte123@gmail.com,
talitha.comaru@iffarroupilha.edu.br, john.soldara@iffarroupilha.edu.br

Abstract. *A problem that has demanded the attention of health professionals is the measurement of pain in newborns as they are unable to communicate through words. Newborn pain measurement scales based on the assessment of behavioral characteristics have been developed decades ago, such as the Neonatal Facial Coding System (NFCS) scale currently used by specialists, and it is based on the assessment of facial expression. On the other hand, with the advancement of facial recognition techniques, such techniques were applied in this project to automatically measure pain in newborns using a database of newborn facial images available on the Internet. To carry out experiments with the proposed techniques, the images were randomly divided into two groups, one for machine learning and the other for validation, since, due to the pandemic, it was not possible to collect new images in hospitals for method validation. Two types of experiments were carried out, in the first one it was measured whether the newborns were suffering or not, and in the second, we tried to measure pain according to the NFCS scale.*

Resumo. *Um problema que tem exigido a atenção de profissionais da saúde é a medição da dor em recém-nascidos uma vez que eles são incapazes de se comunicar através de palavras. Foram desenvolvidas escalas de medição da dor em recém-nascidos baseadas na avaliação de características comportamentais há décadas, como a escala Neonatal Facial Coding System (NFCS), usada atualmente por especialistas, e é baseada na avaliação da expressão facial. Por outro lado, com o avanço das técnicas de reconhecimento facial, tais técnicas foram aplicadas nesse projeto para medir automaticamente a dor em recém-nascidos usando uma base de imagens faciais de recém-nascidos disponível na Internet. Para realizar experimentos com as técnicas propostas, as imagens foram separadas aleatoriamente em dois grupos, um para aprendizado de máquina e outro para validação, uma vez que, devido à pandemia, não foi possível coletar novas imagens em hospitais para validação do método. Dois tipos de experimentos foram*

realizados, onde no primeiro foi medido se os recém-nascidos estavam sofrendo dor ou não, e no segundo, buscou-se medir a dor conforme a escala NFCS.

Palavras-chave: Medição da Dor; Neonatal Facial Coding System; Reconhecimento Facial.

CONTEXTO E MOTIVAÇÃO

Recém-nascidos são incapazes de se comunicar através de palavras, portanto a avaliação da dor nesses pacientes é indireta e geralmente inferida a partir das alterações fisiológicas e de parâmetros comportamentais. A mímica facial é considerada uma avaliação comportamental bastante específica, sensível, e útil na avaliação da dor em recém-nascidos. Porém, em casos em que a necessidade de conforto ou analgesia é imperativa, profissionais de saúde contam apenas com escalas de dor que devem ser preenchidas manualmente, em momentos específicos, sem dispor de um dispositivo contínuo de avaliação da dor utilizando alguma tecnologia avançada.

A possibilidade de se usar um dispositivo eletrônico capaz de realizar a medição automática da dor em recém-nascidos auxiliaria bastante os profissionais da saúde, levando a benefícios visíveis para toda a sociedade. Entretanto, o desenvolvimento de tal dispositivo demanda conhecimentos específicos da área da informática e da área da saúde, tais como processamento de imagens, análise facial, reconhecimento de padrões e escalas de quantificação da dor que sejam reconhecidas e validadas internacionalmente.

A fim de se obter tal dispositivo, pretende-se desenvolver um aplicativo voltado a plataformas móveis, a fim de realizar a medição automática da dor em recém-nascidos, usando como base a imagem facial dos mesmos obtida por meio de uma câmera. O aplicativo fará uso de um método que é proposto neste trabalho para medir a dor em recém-nascidos. Entretanto, devido à dificuldade em se coletar imagens faciais de recém-nascidos em hospitais por

causa da pandemia, foi usada uma base de imagens faciais de recém-nascidos disponível na Internet. Para realizar experimentos com as técnicas usadas, as imagens foram subdivididas aleatoriamente e para os fins de aprendizado de máquina e validação. A seguir, o método proposto é detalhado.

DESCRIÇÃO DA SOLUÇÃO

Atualmente existem diversas escalas internacionais para realizar a medição da dor em recém-nascidos que podem ser empregadas por profissionais da saúde, onde uma das principais é a *Neonatal Facial Coding System* (NFCS) [1], onde parâmetros comportamentais são utilizados em conjunto com indicadores fisiológicos para quantificar a dor. Foi adotada a versão da escala da dor com 10 parâmetros [2], de forma que cada parâmetro da escala é avaliado para compor o score final, que varia de 0 a 10, onde 0 indica não-dor e 10 indica o máximo da dor. Os parâmetros avaliados foram: (1) sobrancelha abaixada, (2) olhos bem fechados, (3) aprofundamento do sulco naso-labial (dobra), (4) lábios abertos, (5) alongamento vertical da boca, (6) alongamento horizontal da boca, (7) língua tensa, (8) tremor de queixo, (9) franzir os lábios, (10) protrusão da língua [2].

Além disso, foi usada no projeto uma base de imagens de recém-nascidos disponível na Internet, que é a *iCOPE (infant Classification Of Pain Expressions)* [3], que consiste de 204 fotografias de 26 recém-nascidos experimentando diferentes estímulos nocivos, incluindo repouso e estímulo de dor aguda. Essa base de imagens é comumente usada em trabalhos de medição automática da dor em recém-nascidos e foi adotada neste trabalho

por ser adequada. As imagens da base foram divididas aleatoriamente em dois grupos: aprendizado de máquina e validação. Além disso, pretendeu-se neste projeto realizar a coleta de imagens de recém-nascidos em hospitais para obter as imagens de validação, entretanto, devido à pandemia se tornou inviável realizar tal tarefa e foram usadas as próprias imagens da base para tal finalidade.

Para se reconhecer uma imagem, primeiramente deve-se realizar a extração de características importantes dessa imagem, chamadas de feições. Através de experimentos, observou-se que as feições que melhor representam as diferentes expressões faciais a partir de imagens faciais de recém-nascidos são os três canais de cor RGB (vermelho, verde e azul) dos pixels que compõem as imagens. Portanto, cada imagem da base de imagens adotada é convertida na forma de um vetor de feições plano composto pela concatenação de todos os pixels de cada canal de cor da imagem inteira. Por exemplo, uma imagem RGB representada por uma matriz de tamanho $10 \times 20 \times 3$, é convertida em um vetor plano com 600 dimensões.

Alternativamente, é proposto nesse projeto usar uma topologia de marcações faciais a fim de se extrair informação de apenas pontos importantes da face que tenham ligação com a expressão facial da dor. Conforme ilustrado na Figura 1, foram marcados manualmente os seguintes pontos faciais nessa ordem: acima do olho esquerdo, abaixo do olho esquerdo, abaixo do olho direito, acima do olho direito, meio dos olhos, nariz, acima do lábio superior e abaixo do lábio inferior. Além desses 8 marcadores, foram adicionados mais 5 marcadores interpolados e igualmente espaçados entre cada par de marcadores originais consecutivos, assim totalizando 43 marcadores na face. Os vetores de feições resultantes foram criados concatenando-se todos os pixels ao redor das vizinhanças quadradas de tamanho 11×11 centrados em cada marcador, usando os canais RGB.

Como resultado, os vetores de feições possuem $43 \times 11 \times 11 \times 3 = 15.609$ dimensões.

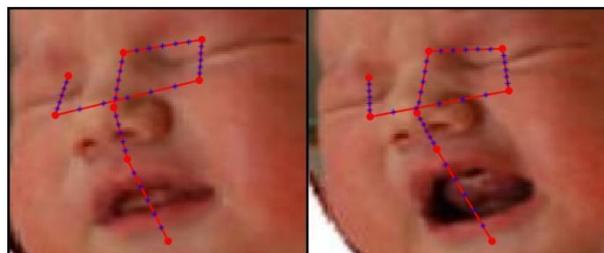


Figura 1. Topologia de marcações adotada exemplificada na base iCOPE [3]

Entretanto, independente da forma que se extraem feições, o processamento direto dos vetores de feições obtidos é inviável devido à sua alta dimensionalidade, o que gera um grande volume de dados para ser processado no aprendizado de máquina, implicando em elevado tempo de processamento e cálculos imprecisos.

Portanto, foi aplicada uma técnica de redução de dimensionalidade adequada de forma a mapear os vetores de feições do espaço original de alta dimensionalidade para um espaço de dimensões reduzidas, onde os vetores resultantes podem ser mais eficientemente manipulados. Para tal fim, foi aplicado o método *Customized Orthogonal Laplacianfaces* [4], que tem por objetivo criar uma projeção linear que preserva ambas a estrutura dos dados e a sua organização em classes de dados. Como resultado, os vetores mapeados para esse espaço possuem uma melhor preservação das classes, levando a uma melhora do processo de classificação das imagens.

Como último passo, é necessária a aplicação de um classificador para classificar as amostras de validação usando como base as amostras reservadas para aprendizado de máquina. O classificador mais amplamente usado na área de reconhecimento de padrões é o classificador do vizinho mais próximo que atribui à uma nova imagem de teste a classe da amostra de treinamento mais semelhante à mesma. Entretanto, através de experimentos, foram

obtidos melhores resultados através da aplicação de Máquinas de Vetores de Suporte com Margens Suaves (SVM) [5], que tem o objetivo de formar uma margem larga de separação entre classes distintas. Essa margem é apoiada nas amostras fronteiriças entre ambas as classes e uma vez que se tenha treinado o classificador durante o aprendizado de máquina, para reconhecer uma nova imagem de teste, a mesma é projetada na direção da margem e a sua classe será dada pelo lado da margem ao qual ela se posicionou. A configuração adotada para o treinamento do classificador SVM é a mesma que é detalhada em [4].

O classificador adotado (SVM) foi aplicado diretamente ao espaço obtido de baixa dimensionalidade, assim permitindo realizar a estimativa da dor em recém-nascidos em duas formas: com duas classes, dor e não-dor; e com a escala NFCS, que corresponde a 11 classes, onde cada classe indica um nível de dor distinto, variando de 0 (não-dor) a 10 (máximo de dor).

PONTOS RELEVANTES DA INOVAÇÃO

Nesse projeto, foi desenvolvido um protótipo do aplicativo móvel com o uso de uma linguagem de programação científica, assim permitindo testar e validar o método proposto de medição da dor que será embutido em dispositivos móveis para serem empregados por enfermeiros, médicos e outros profissionais de saúde na avaliação automática da dor em recém-nascidos.

RESULTADOS OBTIDOS

Classificação em “dor” e “não-dor”

Foram realizados experimentos com imagens faciais inteiras usando o método proposto para medição da dor em recém-nascidos. As 204 imagens da base de imagens adotadas (das quais 144 pertencem à classe “não-dor” e 60 pertencem à classe “dor”) foram divididas aleatoriamente em

dois grupos: aprendizado de máquina com 40 imagens e validação com 20 imagens.

Inicialmente cada imagem RGB foi reduzida ao tamanho de 64x48 pixels e transformada em um vetor de feições concatenando todos os seus pixels. Em seguida, a dimensionalidade dos vetores foi reduzida usando o método COLPP, levando a 17 dimensões finais e, por fim, submetidas ao classificador SVM. Todos os experimentos foram repetidos 10 vezes e a taxa de acerto média obtida foi de 82,25%. Experimentos foram realizados também com as marcações faciais da Figura 1 e a taxa de acerto obtida foi de 87,5%, com uma sensível melhora.

Classificação conforme a escala NFCS

Além disso, experimentos preliminares foram realizados usando a escala de dor NFCS, considerando 11 classes, onde cada classe indica um nível de dor distinto, variando de 0 (não-dor) a 10 (máximo de dor). As imagens da base foram avaliadas por integrantes do projeto usando a escala NFCS, permitindo-se estimar o nível de dor relacionado a cada imagem facial na forma de um número inteiro entre 0 e 10. O mesmo método apresentado anteriormente foi aplicado às imagens faciais de recém-nascidos de forma que para cada classe de dor, duas imagens foram selecionadas para aprendizado de máquina e uma imagem para teste, e a taxa de acerto obtida foi de 26,36%.

CONCLUSÕES

No primeiro conjunto de experimentos realizados, obteve-se resultados satisfatórios em termos de taxa de acerto na classificação das imagens de validação nas classes “dor” e “não-dor”, mas que ainda podem ser melhorados através de melhorias no método proposto.

No segundo conjunto de experimentos, foi realizada a classificação das imagens de

validação contra 11 classes distintas de dor [2], e verificou-se que ainda é necessário melhorar o método proposto para essa modalidade de teste. Como alternativa, poderia ser aplicada outra métrica de classificação que contabilize o quanto se errou ou se acertou no score em cada teste individual, contabilizando a diferença entre o score real e o score atribuído pelo método. Além disso, esse problema poderia ser convertido em um problema de classificação de duas classes, dor e não-dor, ao se considerar dor um score maior ou igual a um determinado limiar.

De forma geral, os experimentos indicaram que é possível realizar a avaliação automática da dor em recém-nascidos e que essa é uma área aberta de pesquisa que pode ser bastante melhorada e pode trazer bastante benefícios à sociedade.

Agradecimentos

Agradecemos a agência de fomento CNPQ e ao IFFAR pelo apoio financeiro ao projeto.

REFERÊNCIAS

- [1]. Grunau R.V.E. Craig K.D. Pain expression in neonates: facial action and cry. *Pain*, 28(3), 1987, pp. 395-410.
- [2]. Zamzmi G, Kasturi R, Goldgof D, Zhi R, Ashmeade T, Sun Y. A Review of Automated Pain Assessment in Infants: Features, Classification Tasks, and Databases. *IEEE Reviews in Biomedical Engineering*, 11, 2018, pp. 77-96.
- [3]. Brahnem S, Chuang C.F, Shih F.Y, Slack M.R. Machine recognition and representation of neonatal facial displays of acute pain. *Artificial Intelligence in Medicine*, 36(3), 2006, pp. 211-22.
- [4]. Soldera J, Behaine C.A.R, and Scharcanski J. Customized Orthogonal Locality Preserving Projections with Soft-Margin Maximization for Face Recognition. *IEEE Transactions on*

Instrumentation and Measurement, 64(9), 2015, pp. 2417-2426.

- [5]. Boser B.E. Guyon I.M. Vapnik, V.N. A training algorithm for optimal margin classifiers. *Proceedings of the fifth annual workshop on Computational learning theory*. 1992, pp. 144.