CENTRO UNIVERSITÁRIO FEI
ENGENHARIA ELÉTRICA

CARLOS ANTÔNIO SCHUMANN NETO
DAVID MASCARENHAS MATANO
RAFAEL OREJAS CHAVES

CLASSIFICAÇÃO AUTOMÁTICA DE DOR EM NEONATOS ATRAVÉS DA SEGMENTAÇÃO DE PARTES DA FACE

SÃO BERNARDO DO CAMPO
2021
CARLOS ANTÔNIO SCHUMANN NETO

DAVID MASCARENHAS MATANO
RAFAEL OREJAS CHAVES

CLASSIFICAÇÃO AUTOMÁTICA DE DOR EM NEONATOS ATRAVÉS DA SEGMENTAÇÃO DE PARTES DA FACE

Trabalho de Conclusão de Curso apresentado como requisito para obtenção de título de Bacharel em Engenharia Elétrica pelo Centro Universitário FEI

Orientador: Dr. Carlos Eduardo Thomaz

Coorientadora: Tatiany Marcondes Heiderich

SÃO BERNARDO DO CAMPO
2021

Schumann Neto , Carlos Antônio .
Classificação automática de dor em neonatos através da segmentação da face / Carlos Antônio Schumann Neto , David Mascarenhas Matano , Rafael Orejas Chaves. São Bernardo do Campo, 2021.
53 f. : il.

Trabalho de Conclusão de Curso - Centro Universitário FEI.
Orientador: Prof. Dr. Carlos Eduardo Thomaz.
Coorientador: Prof. Tatiany Marcondes Heiderich .

1. Neonato. 2. dor. 3. redes neurais convolucionais. 4. detecção facial. I. Mascarenhas Matano , David. II. Orejas Chaves, Rafael . III. Thomaz, Carlos Eduardo , orient. IV. Título.

Elaborada pelo sistema de geração automática de ficha catalográfica da FEI com os dados fornecidos pelo(a) autor(a).

CARLOS ANTÔNIO SCHUMANN NETO

DAVID MASCARENHAS MATANO

RAFAEL OREJAS CHAVES

CLASSIFICAÇÃO AUTOMÁTICA DE DOR EM NEONATOS ATRAVÉS DA SEGMENTAÇÃO DE PARTES DA FACE

Trabalho de Conclusão de Curso apresentado como requisito para obtenção de título de Bacharel em Engenharia Elétrica pelo Centro Universitário FEI

Orientador: Dr. Carlos Eduardo Thomaz
Coorientadora: Tatiany Marcondes Heiderich

Comissão julgadora

_____

Orientador e presidente

_____

Examinador (1)

_____

Examinador (1)

SÃO BERNARDO DO CAMPO

2021

RESUMO

Uma das formas de se avaliar a dor neonatal é através da análise da expressão facial, com avaliação da movimentação das regiões faciais. No entanto, a identificação da dor em recém-nascidos internados em Unidade de Terapia Intensiva Neonatal (UTIN), pode ser dificultada pela presença de aparatos em sua face, que são necessários para a manutenção da vida. Muitos trabalhos estão sendo desenvolvidos para avaliação da expressão facial de dor neonatal, mas o uso de aparatos na região facial inviabiliza a possibilidade de detecção automática da face e sua análise, justamente pela incapacidade de visualizar alguns pontos faciais. Diante disso, este trabalho visa aperfeiçoar o sistema automatizado de identificação de dor em neonatos, enquanto estes encontram-se em leitos ou incubadoras, possibilitando o diagnóstico da face de dor, estando o neonato com ou sem aparatos em sua face. Para isso, foi criado um sistema de segmentação facial utilizando redes neurais convolucionais treinadas e modificadas. Estas redes foram desenvolvidas a partir da utilização de um banco de dados da UNIFESP contendo imagens de neonatos sem artefatos médicos em seus rostos. Como etapas subsequentes há detecção facial, segmentação e extração das áreas de interesse da face e para finalizar foi feito o processo de *data augmentation.* Após essa série de etapas houve a criação do banco de dados para cada segmento de face utilizados para o treinamento das redes neurais convolucionais. Com os bancos de dados criados, as redes conseguiram classificar a dor com uma acurácia aproximada de 71,16% para o seguimento da boca e 76,74% para os seguimentos dos olhos.

Palavras-chave: Neonato, dor, redes neurais convolucionais, detecção facial.

ABSTRACT

One of the ways to assess neonatal pain is through the analysis of facial expression, with assessment of movement of facial regions. However, an identification of pain in newborns hospitalized in the Neonatal Intensive Care Unit (NICU) can be hampered by the presence of devices on their face, which are care for the maintenance of life. Many studies are being carried out to assess the facial expression of neonatal pain, but the use of devices in the facial region makes it impossible to automatically detect the face and its analysis, precisely because of the inability to visualize some facial points. Therefore, this work improves the automated system for identifying pain in neonates, while they appear in beds or incubators, enabling the diagnosis of facial pain, with or without facial devices. For this, a facial treatment system was created using trained and modified convolutional neural networks. These networks were developed using a UNIFESP database containing images of newborns without medical artifacts on their faces. As subsequent steps, there is facial detection, necessary and extraction of the areas of interest on the face and to finalize the data increase process was carried out. After this series of steps, the database was created for each face segment used for training the convolutional neural networks. With the created databases, the networks were able to classify pain with an accuracy of approximately 71.16% for the mouth segment and 76.74% for the eye segments.

Keywords: Neonate, pain, convolutional neural networks, face detection.

LISTA DE ILUSTRAÇÕES

Figura 1 - Três Diagramas de Métodos de Análise de Dor em Neonatos 17
Figura 2 - Exemplo da arquitetura de uma rede neural convolucional (LeNet-5) 19
Figura 3 - Exemplo do processo de convolução 19
Figura 4 - Funcionamento do max pooling. 20
Figura 5 - Disposição das câmeras para a captura de imagens do neonato. . 23
Figura 6 - Fluxograma das etapas do projeto. 24
Figura 7 - Estrutura da Rede e Perdas Multitarefas. 26
Figura 8 - Exemplo de retorno das coordenadas pelo *RetinaFace*. 27
Figura 9 - Exemplo da extração das áreas de interesse 27
Figura 10 - Exemplo do processo de convolução. 29
Figura 11 - Comparação com o Estado-da-Arte. 31
Figura 12 - Gráfico de performance das redes 32
Figura 13 - Taxa de erro das redes 35
Figura 14 - Representação da criação dos *datasets.* 38
Figura 15 - Equação da dimensão das imagens geradas. 39
Figura 16 - Gráfico da taxa de acerto das imagens da boca (Batch size 1) ... 41
Figura 17 - Gráfico de erro para a boca (Batch size 1) 41
Figura 18 - Gráfico da taxa de acerto das imagens do olho esquerdo (Batch size 1) 42
Figura 19 - Gráfico de erro para o olho esquerdo (Batch size 1) 42
Figura 20 - Gráfico da taxa de acerto das imagens do olho direito (Batch size 1) 43
Figura 21 - Gráfico de erro para o olho direito (Batch size 1) 43
Figura 22 - Gráfico da taxa de acerto das imagens da boca (Batch size 16) . 45
Figura 23 - Gráfico de erro para a boca (Batch size 16) 45
Figura 24 - Gráfico da taxa de acerto das imagens do olho esquerdo (Batch size 16) 46
Figura 25 - Gráfico de erro para o olho esquerdo (Batch size 16) 46
Figura 26 - Gráfico da taxa de acerto das imagens do olho direito (Batch size 16) 47
Figura 27 - Gráfico de erro para o olho direito (Batch size 16) 47

LISTA DE TABELAS

Tabela 1 – Escala de dor para Recém-Nascidos (NIPS)..... 12
Tabela 2 - *Premature Infant Pain Profile* 16
Tabela 3 - Sistema de Codificação da Atividade Facial Neonatal (*Neonatal Facial Coding System* -NFCS)..... 16
Tabela 4 - Camads de Lenet 5..... 19
Tabela 5 – *Datasets* após cada estágio de processamento. 30
Tabela 6 - Arquiteturas e suas respectivas acurácias. 31
Tabela 7 – Comparativo entre os *Datasets*. 32
Tabela 8 – Etapas de construção dos *datasets.* 37
Tabela 9 - Valores dos parâmetros utilizados para o primeiro treino. 40
Tabela 10 - Valores dos parâmetros utilizados para o segundo treino. 44

LISTA DE ABREVIATURAS

AM	Aprendizado de Máquina
AP	Aprendizado Profundo
NB	*Newborn*
N-CNN	*Neonatal Convolutional Neural Network*
PIPP	*Premature Infant Pain Profile*
RN	Recém-Nascido
RNA	Rede Neural Artificial
RNC	Rede Neural Convolucional
UTIN	Unidade de Terapia Intensiva Neonatal

Sumário

1. INTRODUÇÃO 11
1.1 OBJETIVO 13
1.2 ESTRUTURA DESTA DISSERTAÇÃO 13
2. TRABALHOS RELACIONADOS 14
2.1 DOR EM NEONATOS E SUAS CONSEQUÊNCIAS 14
2.2 AVALIAÇÃO DA DOR 15
2.3 ANÁLISE AUTOMÁTICA DA DOR 17
2.4 REDES NEURAIS CONVOLUCIONAIS 18
3. MATERIAIS E METODOLOGIA 22
3.1 SOFTWARES 22
3.2 BANCO DE DADOS 22
3.3 METODOLOGIA 24
3.3.1 Detecção facial 25
3.3.2 Extração das Áreas de Interesse 26
3.3.3 *Data Augmentation* 27
3.3.4 Classificação de imagens 28
3.4 Arquiteturas de Redes Neurais Convolucionais 29
3.4.1 VGG FACE 30
3.4.2 VGG FACE 2 31
3.4.3 LENET5: 33
3.4.4 *DataSet*: *The modified NIST set* 34
4. Resultados 36
4.1 PROCEDIMENTOS 36
4.2 RESULTADOS 40
5. Conclusão 49
Referência Bibliográfica 49

1. INTRODUÇÃO

Durante muitos anos, pensava-se que recém-nascidos não sentiam ou lembravam da dor como adultos e crianças mais velhas, devido ao fato de não terem o sistema nervoso totalmente desenvolvido. Atualmente essa visão se tornou ultrapassada e diversos estudos comprovaram que até mesmo os fetos, a partir de 24 semanas de gestação, sentem dor (ANAN D and C RAIG 1996) (Anand, et al. 1987) (LS, CS and B 2000).

De acordo com os estudos realizados por Anand e Craig (1996), Anand e Carr (1989), *Craig* et al. (1993), os sistemas anatômicos, sensoriais e neuroquímicos já estão suficientemente desenvolvidos para permitir que os neonatos tenham a percepção de dor e apresentam respostas cardiorrespiratórias, hormonais e comportamentais. Estudos também apontam que a exposição excessiva a dor pode acarretar problemas futuros, devido as alterações morfológicas geradas com o desenvolvimento da criança. Alguns dos problemas gerados estão conectados ao sistema nervoso central, impactando especialmente nas habilidades viso-motoras. (Portal de Boas Práticas em Saúde da Mulher, da Criança e do Adolescente 2018).

O uso excessivo de analgésicos também pode provocar malefícios para o recém-nascido, tais como complicações de curto prazo e sequelas fisiológicas, comportamentais e cognitivas de longo prazo. Alguns exemplos de problemas gerados a longo prazo são o processamento alterado da dor, transtorno de déficit de atenção e diminuição da integração motora. (Hall and Anand 2015)

Diante destas complicações, e da necessidade de avaliar a dor de forma precisa, diversos estudos sobre a avaliação de dor neonatal foram feitos e foram criadas diversas maneiras de medir a mesma, umas delas é a avaliação comportamental. Uma das escalas criadas para esta avaliação é a Escala de Dor do Neonato (Tabela 1) (NIPS – *Neonatal Infant Pain Scale*).

Tabela 1 – Escala de dor para Recém-Nascidos (NIPS)

Tabela 1

Escala de Dor para Recém-Nascidos=Neonatal Infant Pain Scale (NIPS).

Indicador	0 ponto	1 ponto	2 pontos
Expressão facial	Relaxada	Contraída	---
Choro	Ausente	Resmungos	Vigoroso
Respiração	Regular	Diferente da basal	---
Braços	Relaxados	Fletidos/Estendidos	---
Pernas	Relaxadas	Fletidas/Estendidas	---
Estado de Alerta	Dormindo e/ou Calmo	Agitado e/ou Irritado	---

Presença de dor:>3 pontos

Fonte: (Lawrence, et al. 1993)

Sabendo que a avaliação de dor pode ser feita através da análise da expressão facial e corporal, uma forma de tentar auxiliar este diagnóstico, sem o viés humano, é através do desenvolvimento de uma tecnologia capaz de identificar a dor de forma automática.

Muitos trabalhos estão sendo desenvolvidos para avaliação da expressão facial de dor neonatal de forma automatizada, mas o uso de artefato, tais como máscaras de respiração ou tampões nos olhos, impossibilitam a análise do rosto como um todo. Uma hipótese, para amenizar este problema, seria segmentar a face para poder identificar a dor mesmo quando há algum aparato obstruindo a face e mesmo diante desta situação, ser possível avaliar a dor.

1.1 OBJETIVO

Esta monografia tem por objetivo, desenvolver uma ferramenta para avaliar a expressão facial de dor do recém-nascido, a partir da análise de regiões faciais neonatais segmentadas.

1.2 ESTRUTURA DESTA DISSERTAÇÃO

Esta dissertação está estruturada em 5 capítulos. No capítulo seguinte estão os trabalhos relacionados utilizados para embasar os assuntos abordados durante a monografia, tais como métodos computacionais e avaliação automática da dor por meio de redes neurais. No Capítulo 3 são apresentados os materiais e as metodologias utilizadas para segmentação da face, análise e reconhecimento de dor nos neonatos. No Capítulo 4 são apresentados os dados experimentais e os resultados obtidos durante as análises. Para encerrar, no capítulo 5 estão as considerações finais e a conclusão.

2. TRABALHOS RELACIONADOS

Neste capítulo, são abordados os temas relacionados à temática de reconhecimento e avaliação da dor em neonatos e métodos computacionais pertinentes ao desenvolvimento do trabalho.

2.1 DOR EM NEONATOS E SUAS CONSEQUÊNCIAS

Durante muitos anos acreditou-se que recém-nascidos ou neonatos, quando expostos a experimentos dolorosos, não sentiam ou lembravam da dor. Essa crença foi estabelecida pelo simples fato de que os lactantes não são capazes de se comunicar verbalmente e explicar suas emoções. Acreditava-se que eles sentiam dor em menor escala e em pouco tempo iriam esquecer do evento doloroso (ANAN D and C RAIG 1996). Essa crença se tornou uma inverdade juntamente com o avanço dos estudos da saúde relacionados ao assunto. Dentre as descobertas feitas, foi comprovado que o desenvolvimento anatômico para percepção de dor inicia-se durante o período de gestação, confirmando que RN são capazes de sentirem dor (ANAN D e C RAIG 1996). A primeira parte a ser desenvolvida são os receptores sensoriais cutâneos, que se inicia na 7ª semana de gestação e vai se alastrando primeiramente pelo rosto, seguida para as palmas das mãos e atinge as solas dos pés na 11ª semana. Nas semanas subsequentes o feto continua seu desenvolvimento e torna-se capaz de sentir dor entre a 20ª e 24ª semanas de gestação.

O aprofundamento do tema dor em neonatos é de vital relevância pois os RN criticamente doentes, já que eles são expostos a uma média de 50 a 150 procedimentos dolorosos nas primeiras semanas de vida quando internados em UTIN (Unidade de Terapia Intensiva Neonatal). Os neonatos, quando nascem com menos de mil gramas, são expostos a uma quantidade ainda maior de processos dolorosos, em torno de 500 procedimentos durante a primeira semana de vida. (Grunau and Craig 1986).

Para a avaliação da dor neonatal é de extrema importância que seja levado em consideração indicadores fisiológicos e comportamentais. As variáveis fisiológicas incluem frequência cardíaca, frequência respiratória, pressão arterial, saturação de oxigênio, pressão transcutânea de oxigênio e de dióxido de carbono e dosagens hormonais, ligadas à resposta endócrino-metabólica de estresse, entre outras. Dentre

as principais reações comportamentais estão o choro, atividade motora e mímica facial (Grunau and Craig 1986).

Dentre os problemas gerados pela dor em excesso, é possível observar alguns problemas relacionados à emoção e à dificuldade em lidar com estresse. Dentre estes problemas alguns são imediatos, como hipersensibilidade a dor e distúrbio de sono. Outros problemas ocorrem a longo prazo, como dificuldade de aprendizado e problemas comportamentais (Mitchell and Boss 2002).

A forma mais eficiente para avaliação da dor e sua intensidade se dá através da combinação de reações comportamentais e indicadores fisiológicos. Com base nisso, tem-se uma resposta segura a respeito do estado do neonato (Grunau e Craig 1986).

2.2 AVALIAÇÃO DA DOR

Durante muitos anos foram desenvolvidas técnicas de reconhecimento da dor para que esta fosse identificada de forma rápida e eficiente. Dentre os métodos possíveis, temos métodos unidimensionais e multidimensionais. Um método multidimensional seria um algoritmo que visa avaliar os neonatos através de múltiplos parâmetros, como expressões da face e volume do choro ou atividade motora e frequência respiratória. Já os métodos unidimensionais seriam aqueles focados em uma única fonte de dados, como somente a expressão facial (Guinsburg 1999).

Dentre os vários métodos multidimensionais existentes podemos destacar a escala *Neonatal Infant Pain Scale* (NIPS) (Tabela 1) que possui 4 indicadores comportamentais e fisiológico; e a escala *Premature Infant Pain Profile* (PIPP) (Tabela 2).

Tabela 2 - *Premature Infant Pain Profile*

	Indicadores	0	1	2	3
	IG (sem.)	≥ 36	32 – 356/7	28 – 316/7	< 28
Observar RN 15 seg. Anotar FC / Sat. O_2 basais	**Estado de alerta**	ativo acordado olho aberto movimentos faciais +	quieto acordado olho aberto sem mímica facial	ativo dormindo olho fechado movimentos faciais +	quieto dormindo olho fechado sem mímica facial
Observar RN 30 seg.	**FC Máxima**	↑0 - 4 bpm	↑5 - 14 bpm	↑15 - 24 bpm	↑≥ 25 bpm
	SO_2 Mínima	↓0 - 2,4%	↓2,5 - 4,9%	↓5,0 - 7,4%	↓≥ 7,5%
	Testa franzida	ausente	mínimo	moderado	máxima
	Olhos esprimidos	ausente	mínimo	moderado	máxima
	Sulco naso-labial	ausente	mínimo	moderado	máxima

Fonte: (Stevens, et al. 1996)

Para o desenvolvimento deste trabalho foi escolhido utilizar a escala *Neonatal Facial Coding Sistem* (NFCS) (Tabela 3), devido sua eficiência e praticidade por apresentar caráter unidimensional, avaliando apenas a expressão facial do RN (Grunau e Craig 1986).

Tabela 3 - Sistema de Codificação da Atividade Facial Neonatal (*Neonatal Facial Coding System* - NFCS)

Movimento Facial	0 Pontos	1 Ponto
Fronte saliente	Ausente	Presente
Fenda palpebral estreitada	Ausente	Presente
Sulco naso-labial aprofundado	Ausente	Presente
Boca aberta	Ausente	Presente
Boca estirada (horizontal ou vertical)	Ausente	Presente
Língua tensa	Ausente	Presente
Protrusão da língua	Ausente	Presente
Tremor de queixo	Ausente	Presente

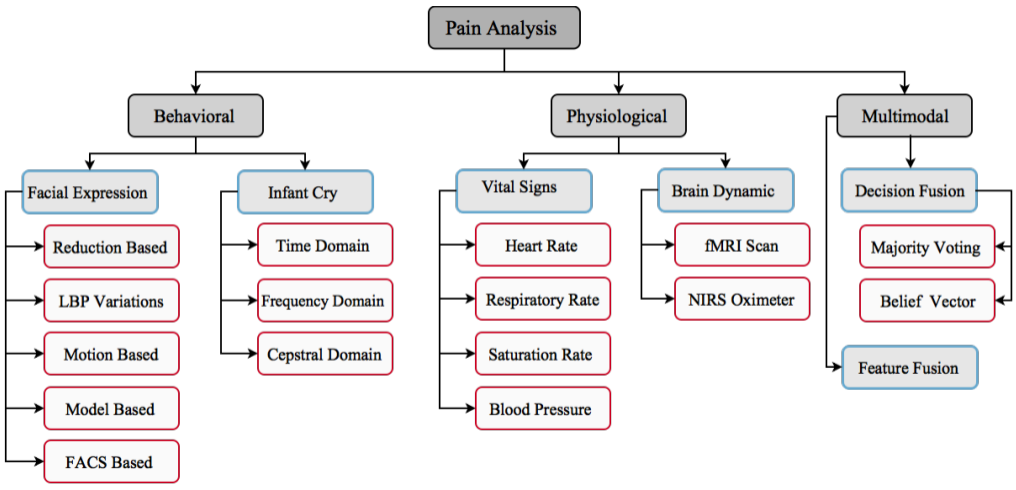
Fonte (Grunau e Craig 1986)

2.3 ANÁLISE AUTOMÁTICA DA DOR

Durante os últimos anos foi possível ver a implementação do aprendizado de máquina (AM) em diversos âmbitos e isso se deve as diversas usabilidades e benefícios que esta tecnologia trás. Ao se aprofundar na área da saúde por exemplo, podemos ver que diversos estudos a respeito do reconhecimento de dor foram e estão sendo feitos utilizando o AM. Esse efeito torna-se possível devido a capacidade de interpretar as expressões faciais, assim podendo determinar a resposta. (Werner, AL-HAMADI and NIESE 2014).

Embora haja o avanço nas pesquisas relacionadas a detecção de dor, a área focada em detecção em neonatos ainda não é tão explorada e devido a alguns fatores específicos, tais como a impossibilidade de aplicar os mesmos meios de detecção que são feitos em adultos pois os dimensionamentos dos rostos são diferentes e muitas vezes os RN não cooperam como desejado. A partir disso a Dra. Ghada Zamzmi desenvolveu um método de análise da dor em neonatos mostrado na figura 1, dividido em 3 parâmetros: comportamental, fisiológico e multimodal (Zamzmi, Paul, et al., Pain Assessment From Facial Expression: Neonatal Convolutional Neural Network (N-CNN) 2019).

Figura 1 - Três Diagramas de Métodos de Análise de Dor em Neonatos



Fonte: (Zamzmi, Paul, et al., A Review of Automated Pain Assessment in Infants 2017)

Com base nisso, Zamzmi junto a outros pesquisadores, desenvolveram a partir de um *Survey,* um tipo específico de rede neural, focada no uso de imagens para

análise de reconhecimento da dor no neonato. Esta rede foi batizada de *Neonatal Convolutional Neural Network* (N-CNN). (Zamzmi, Paul, et al., Pain Assessment From Facial Expression: Neonatal Convolutional Neural Network (N-CNN) 2019).

2.4 REDES NEURAIS CONVOLUCIONAIS

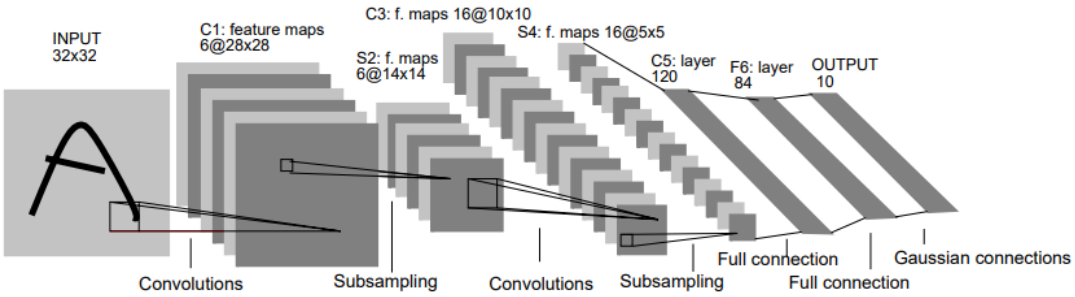
Rede neural convolucional (RNC) é um tipo de rede neural artificial inicialmente proposta por *LeCun* (Lecun, et al. 1998), que vem sendo bastante utilizada em visão computacional para classificação de imagens. Este modelo de rede neural artificial possui grande eficácia na redução da sensibilidade de distorção, rotação e translação, mas necessita de uma grande quantidade de dados rotulados para alcançar bons resultados.

Existem diversos modelos de RNC, porém estes modelos são comumente constituídos por uma camada de convolução responsáveis pela extração de características da imagem, camadas de *pooling* para diminuir a dimensão da imagem e uma camada totalmente conectada no final da sequência responsável pela classificação. Na camada de convolução, cada pixel de um filtro de pesos W (filtro denominado *kernel*) é multiplicado pelos pixels da imagem de entrada e então feita a somatória dos produtos obtidos, resultando em um novo pixel da nova imagem gerada, denominada *feature map*. Após o primeiro pixel do *feature map* ter sido calculado, o *kernel* desloca-se N pixels para a direita e ao chegar ao final da imagem, volta ao início e desloca-se N pixels para baixo. O valor de N é chamado *stride* e indica a movimentação do filtro convolucional na imagem. Ao final do processo de convolução, é obtido um *feature map* com tamanho calculado pela equação:

$$T = \frac{Im-k}{s} + 1 \quad (1)$$

onde T é o tamanho do *feature map* resultante, Im o tamanho da imagem de entrada, k o tamanho do *kernel* e s o *stride*. A arquitetura da rede convolucional Lenet5 é apresentada na Figura 2 e na Tabela 4, de forma um pouco mais detalhada.

Figura 2 - Exemplo da arquitetura de uma rede neural convolucional (LeNet-5)



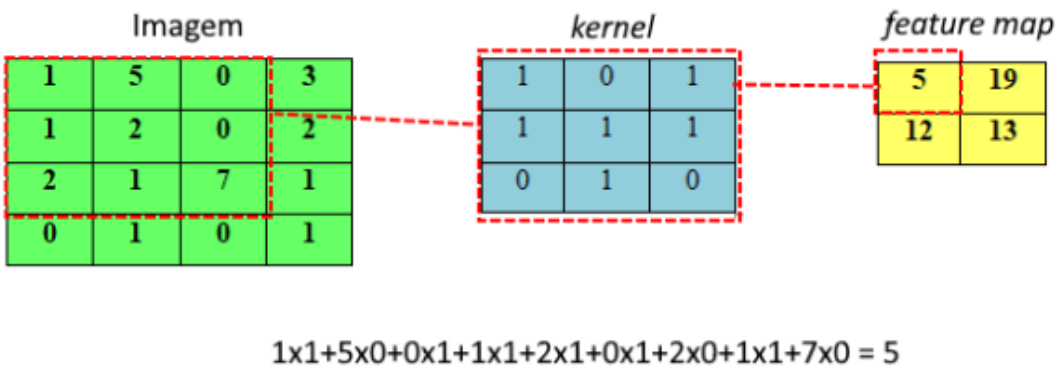
Fonte: (Lecun, et al. 1998)

Tabela 4 - Camadas de Lenet 5.

S.No	Layers	Output Shape (Height, Width, Channels)
1	Input Layer	32 x 32 x 1
2	Conv2d [6 Filters of size = 5x5, stride = 1, padding = 0]	28 x 28 x 6
3	Average Pooling [stride = 2, padding = 0]	14 x 14 x 6
4	Conv2d [16 Filters of size = 5x5, stride = 1, padding = 0]	10 x 10 x 16
5	Average Pooling [stride = 2, padding = 0]	5 x 5 x 16
6	Conv2d [120 Filters of size = 5x5, stride = 1, padding = 0]	1 x 1 x 120
7	Linear1 Layer	120
8	Linear2 Layer	84
9	Final Linear Layer	10

Fonte: (Balakrishnakumar 2020)

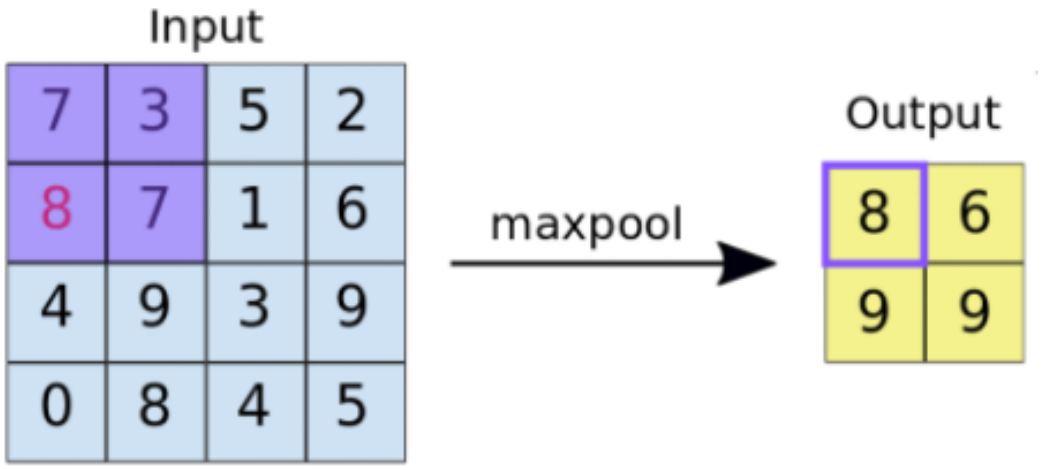
Figura 3 - Exemplo do processo de convolução



Fonte: Autor

Percebe-se que após a imagem passar pelo *kernel* seu tamanho é reduzido, mas é possível alterar o tamanho do *feature map* usando uma técnica chamada de *padding* ou *zero-padding* que adiciona pixels com valores iguais a zero em torno a imagem de entrada antes da filtragem. Após a filtragem normalmente adiciona-se uma função de ativação do tipo não-linear, como a ReLU, para evitar problemas não-lineares. Em seguida é feito o processo de *pooling* para alcançar certo nível de invariância ao deslocamento reduzindo o tamanho da *feature map.* A Figura 4 exemplifica o processo de *max pooling*, que consiste em um algoritmo que seleciona o *pixel* de maior valor dentro de uma "janela" de *pixels* e o adiciona a nova imagem. O filtro da Figura 4 possui *stride* igual à dois.

Figura 4 - Funcionamento do *max pooling*.



Fonte: (Alves 2018)

Finalizadas as etapas de convolução, é feito o procedimento de *flatten* na imagem resultante, que consiste em transformar a imagem multidimensional em unidimensional armazenando seus pixels em um vetor. A última etapa do processo consiste na interpretação das características extraídas das etapas anteriores. Para isto, usa-se uma rede neural totalmente conectada e uma camada final do tipo *softmax.* O número de neurônios da camada *softmax* depende do número de classes finais da classificação e seus valores são representados pela equação abaixo:

$$yj = \frac{e^{y(j)}}{\sum_{i=1}^{n} e^{y(i)}} \quad (2)$$

onde a saída y do neurônio j passa a ser o valor da mesma dividido pelo somatório de todas as saídas dos neurônios dessa camada (EBERMAM, E., KROHLING, R., 2018).

O aprendizado da RNC é feito pelo *back propagation*, que usa a descida do gradiente na atualização dos erros. O cálculo do erro mais usado em problemas com duas ou mais classes de classificação é a função de entropia cruzada:

$$Loss = -\sum_{i=1}^{n} yi \log(y^{*}i) \tag{3}$$

onde $y^{*}i$ é o i-ésimo valor escalar da saída do modelo, yi o valor alvo e n o número de elementos na saída (CATEGORICAL CROSSENTROPY et al., 2021). A atualização dos pesos é feita conforme as equações a seguir:

$$\Delta\omega_i^t = \eta \frac{\delta C}{\delta \omega_i^t} \tag{4}$$

$$\omega_i^{t+1} = \omega_i^t - \Delta\omega_i^t \tag{5}$$

onde ω_i^t é um peso i no instante igual a t (atual), $\Delta\omega_i^t$ a variação do peso no instante t e η taxa de aprendizagem. Durante a etapa de treinamento a rede pode se adaptar muito bem aos dados inseridos, mas possuir baixa eficácia a novos dados adicionados. Para evitar este efeito (*overfitting*) é usada a técnica de *dropout*, que desativa uma porcentagem de neurônios da rede durante o treinamento.

3. MATERIAIS E METODOLOGIA

Este capítulo detalha a metodologia e as ferramentas utilizadas no desenvolvimento do projeto. Primeiramente, serão apresentados os softwares e ambientes de desenvolvimento utilizados na construção do projeto. Em seguida será mostrado o banco de imagens e os métodos computacionais que constituem o projeto.

3.1 SOFTWARES

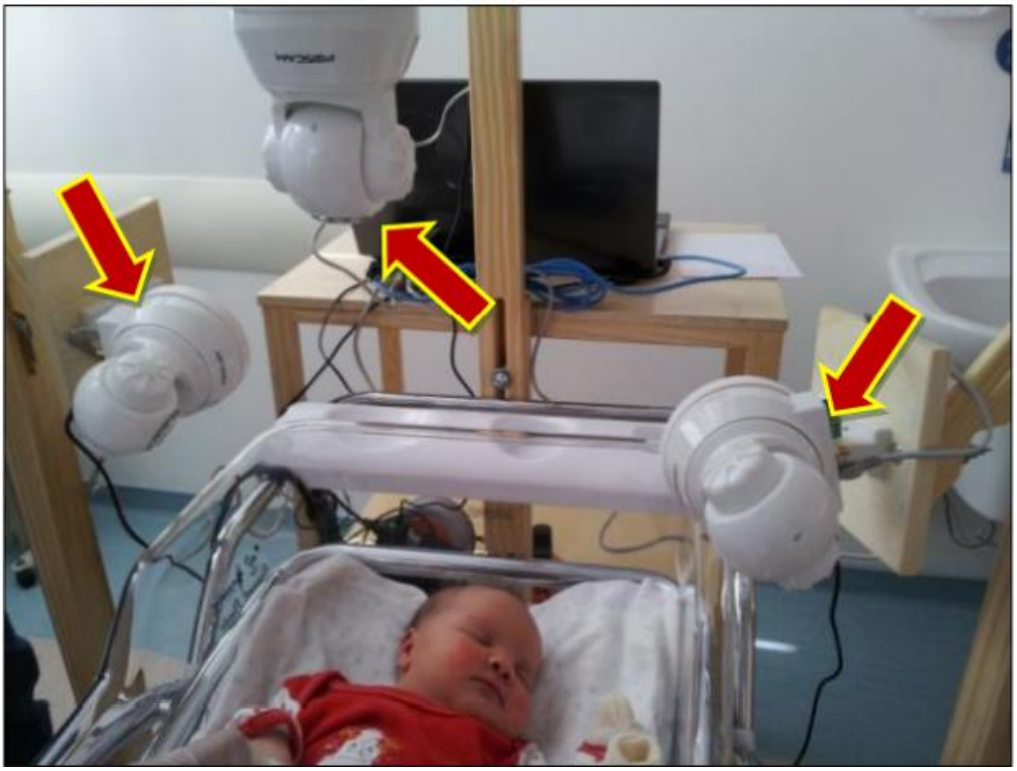
O desenvolvimento do projeto foi realizado no Google Colab que é um ambiente de *Jupyter Notebook* executado na nuvem do Google. A linguagem de programação utilizada no desenvolvimento do projeto foi *Python*3 e as bibliotecas usadas foram *OpenCV* e *Pytorch*, ambas amplamente utilizadas em visão computacional e *Machine Learning*.

3.2 BANCO DE DADOS

O banco de imagens utilizado no modelo computacional foi o banco da UNIFESP (Heiderich, Guinsburg and Leslie 2013), constituído por 360 imagens coloridas de recém-nascidos entre 34 e 41 semanas de idade e entre 24 e 168 horas de vida. Os neonatos participaram da pesquisa sem nenhuma analgesia, por solicitação médica com a autorização dos pais ou responsáveis e com a permissão do uso de imagem do comitê de ética, sob o número CCAE: 66743417.0.0000.5505 e o número de parecer 2.035.113 (HEIDERICH, LESLIE and GUINSBURG 2015)

A construção do banco de imagens com as faces de neonatos foi feita utilizando três câmeras digitais dispostas sobre o leito do recém-nascido. Foram feitas filmagens enquanto o neonato era submetido a procedimentos potencialmente dolorosos como punção venosa, capilar ou injeção intramuscular, com dez minutos de duração e tempo de captura das imagens de três segundos. (HEIDERICH, LESLIE e GUINSBURG 2015)

Figura 5 - Disposição das câmeras para a captura de imagens do neonato.



Fonte: Imagem retirada de Heiderich, Guinsburg e Leslie (2013).

As imagens capturadas foram denominadas de acordo com as etapas dos procedimentos realizados e separadas em 3 categorias, que são situação real sem dor (adquirida antes do procedimento potencialmente doloroso), situação real com dor (adquirida durante o procedimento potencialmente doloroso) e recuperação (após um intervalo de tempo do procedimento) (HEIDERICH, LESLIE e GUINSBURG 2015).

Após a captura das imagens, foram selecionadas 12 imagens faciais para cada um dos 30 recém-nascidos com a ajuda de profissionais da saúde. Dentre as 360 imagens faciais resultantes, 138 foram adquiridas durante o repouso, 30 durante o procedimento doloroso e 192 imagens foram colhidas 10 minutos após o procedimento. A partir da análise das imagens pelos profissionais da saúde usando como parâmetro escala de dor NFCS, as imagens foram classificadas em “com dor”, “com desconforto” ou “sem dor”. Posteriormente, as classificações “com dor” e “com desconforto” foram agrupadas em um único campo “com dor” (HEIDERICH, LESLIE e GUINSBURG 2015).

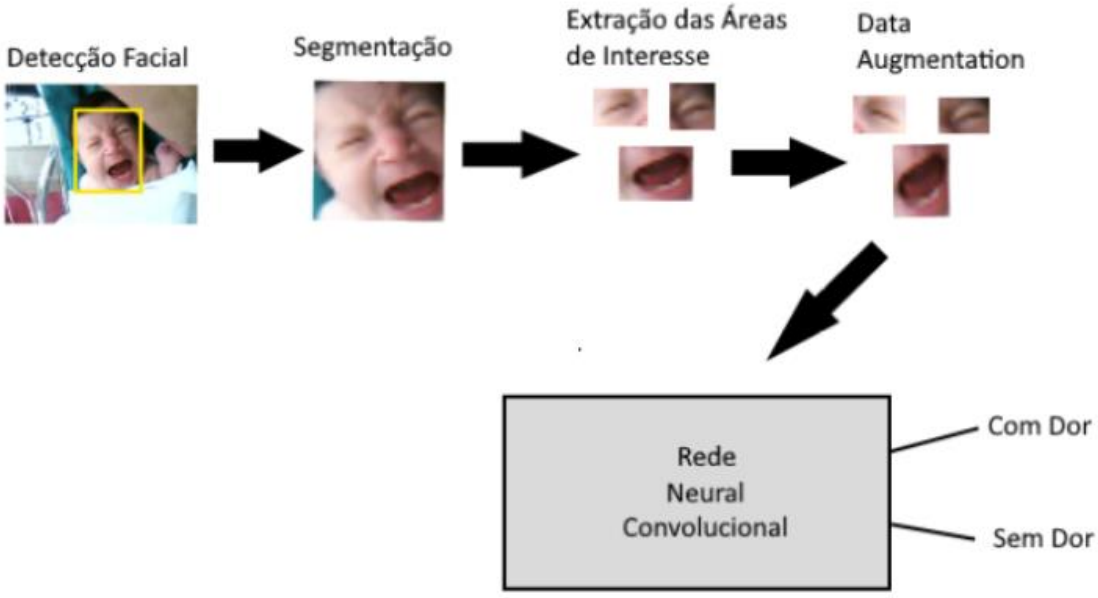
As 360 imagens geradas de 450x233 pixels foram adquiridas em situações reais de internação e não foram submetidas a algoritmos de extração de regiões de interesse. Devido a esta realidade, não foram eliminadas as chances de sobreposição

das faces dos recém-nascidos por algum equipamento hospitalar ou até mesmo as mãos do neonato (HEIDERICH, LESLIE e GUINSBURG 2015).

3.3 METODOLOGIA

Nesta seção serão apresentados os métodos utilizados na construção de cada etapa do projeto experimental, que são: detecção facial do recém-nascido, segmentação da face, extração das regiões de interesse como os olhos e a boca do neonato, *data augmentation* para aumento da base de dados e a classificação das imagens como “com dor” e “sem dor” para partes isoladas do rosto. A imagem a seguir apresenta o fluxograma das etapas seguidas.

Figura 6 - Fluxograma das etapas do projeto.



Fonte: Autor

3.3.1 Detecção facial

A detecção facial é um método de visão computacional que busca em uma imagem características relacionadas a face humana e retorna as coordenadas da caixa delimitadora que possivelmente contém o rosto. Neste projeto, a etapa de detecção facial foi realizada utilizando o algoritmo de detecção facial denominado *RetinaFace*.

RetinaFace (DENG et al 2020) é um algoritmo computacional de detecção de face de estágio único que foi aprimorado pelos autores usando métodos de localização de face densa, usando aprendizagem multitarefa auto supervisionada e extra supervisionada conjunta. A localização da face realizada pelo algoritmo inclui detecção da face, alinhamento facial, reconstrução 3D do rosto e segmentação.

Visando a minimização do erro de caixa de âncora i (caixa que contém informações da imagem detectada, se é ou não uma face e retorna as coordenadas da região que a contém), os autores utilizaram o erro de multitarefas. O erro de multitarefas é representado na equação abaixo:

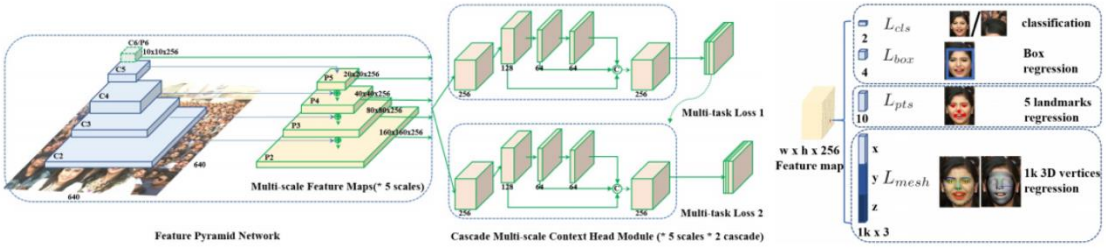
$$L = L_{cls}(p_i, p_i^*) + \lambda_1 p_i^* L_{box}(t_i, t_i^*) + \lambda_2 p_i^* L_{pts}(l_i, l_i^*) + \lambda_3 p_i^* L_{pixel} \quad (6)$$

Onde o termo $L_{cls}(p_i, p_i^*)$ é o erro de classificação, L_{cls} a função softmax para classes binárias, p_i a probabilidade de a caixa de âncora ser uma face e p_i^* recebe 1 ou 0, dependendo dos valores de i serem positivos ou negativos. Os valores de λ_1, λ_2 e λ_3 são erros que recebem como parâmetros de balanceamento os valores 0,25, 0,1 e 0,01. $L_{box}(t_i, t_i^*)$ representa a perda da regressão da região facial, onde os parâmetros t_i e t_i^* indicam as coordenadas da caixa prevista e a real região normalizada associada ao i positivo e são representados por $\{t_x, t_y, t_w, t_h\}i$ e $\{t_x^*, t_y^*, t_w^*, t_h^*\}i$, respectivamente. O erro de regressão dos pontos faciais indicado por $L_{pts}(l_i, l_i^*)$, tem $l_i = \{l_{x1}, l_{y1}, \dots, l_{x5}, l_{y5}\}$ e $l_i^* = \{l_{x1}^*, l_{y1}^*, \dots, l_{x5}^*, l_{y5}^*\}i$ como parâmetros e representam cinco marcos faciais, para os valores previstos e a para a real região normalizada associada ao i positivo, respectivamente. L_{pixel} (erro da regressão densa) é representado na Equação (7).

$$L_{pixel} = \frac{1}{W*H} \sum_i^W \quad \sum_j^H \quad ||\mathcal{R}(\mathcal{D}_{Pst}, P_{cam}, P_{ill})i, j - I_{i,j}^*|| \, 1 \quad (7)$$

onde, $P_{st}, P_{cam}, P_{ill}, W$e H são parâmetros preditos de forma e textura (para projetar malha colorida $\mathcal{D}_{Pst}$), posição e distância focal da câmera, iluminação, largura e altura do recorte $I^*_{i,j}$, respectivamente.

Figura 7 - Estrutura da Rede e Perdas Multitarefas.



Fonte: (DENG et al 2020)

3.3.2 Extração das Áreas de Interesse

Feita a detecção facial nas imagens e recortada a região facial para posterior análise, foi feita a extração das regiões dos olhos e boca levando em consideração a possibilidade de alguma dessas regiões de interesse estar coberta por um adereço hospitalar ou as mãos do recém-nascido. Cada imagem recortada (olho esquerdo, olho direito e boca) foi atribuída ao neonato de origem e analisadas posteriormente pelo modelo computacional, sendo classificadas como olho direito com ou sem dor, olho esquerdo com ou sem dor e boca com ou sem dor.

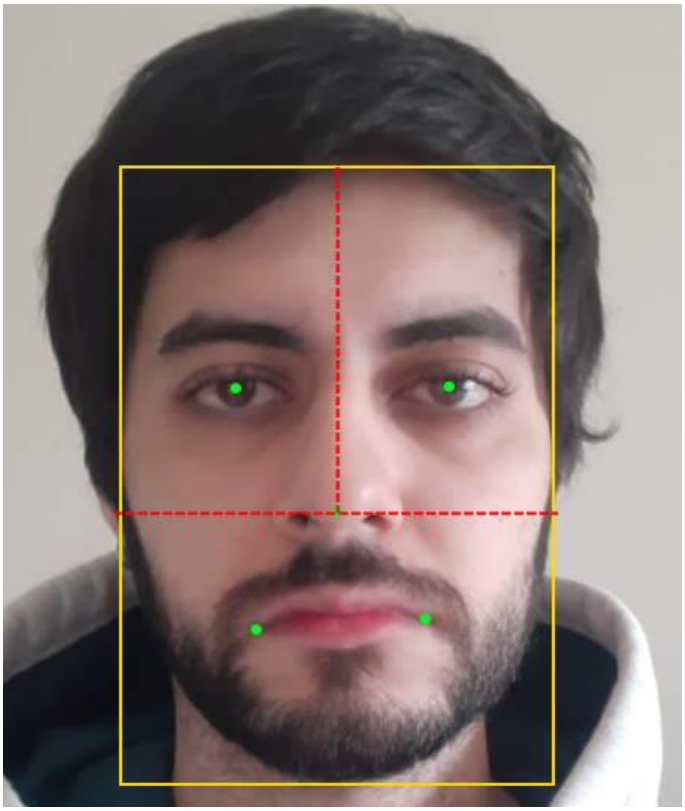
A extração das regiões está ilustrada na Figura (9). Após o algoritmo do detector facial retornar as coordenadas dos olhos, boca, nariz e delimitação da face, o recorte foi feito passando pelo ponto do nariz verticalmente e horizontalmente. As 3 imagens resultantes desse recorte (olho esquerdo, olho direito e boca) foram classificadas como com dor ou sem dor.

Figura 8 - Exemplo de retorno das coordenadas pelo *RetinaFace*.

```
{'face_1': {'score': 0.999904990196228,
  'facial_area': [398, 153, 578, 367],
  'landmarks': {'right_eye': [425.52197, 227.16914],
   'left_eye': [492.6687, 215.231],
   'nose': [431.51407, 233.26247],
   'mouth_right': [427.53015, 305.22992],
   'mouth_left': [481.80832, 295.74567]}}}
```

Fonte: Autor

Figura 9 - Exemplo da extração das áreas de interesse.



Fonte: Autor

3.3.3 *Data Augmentation*

Após realizar a extração das áreas de interesse das imagens dos recém-nascidos, foi aplicado o processo de *data augmentation*. Este processo consiste em aumentar a quantidade de imagens que foram utilizadas no treinamento do modelo computacional criando cópias e as modificando de diferentes formas, como inversão horizontal ou vertical, rotação, mudança de cor ou recorte da imagem.

A técnica de *data augmentation* foi utilizada visando diminuir o *overfitting*, que de forma reduzida podemos descrevê-lo como um termo utilizado em *machine learning* e estatística para dizer que um modelo se arranja muito bem aos dados que foram usados na etapa de treinamento, porém os "decorou" e apresenta dificuldades em prever dados que se diferem do conjunto de dados de treinamento. A utilização do *data augmentation* foi feita através do *framework Tensorflow*, pois este possui ferramentas que facilitaram o processo de desenvolvimento do projeto.

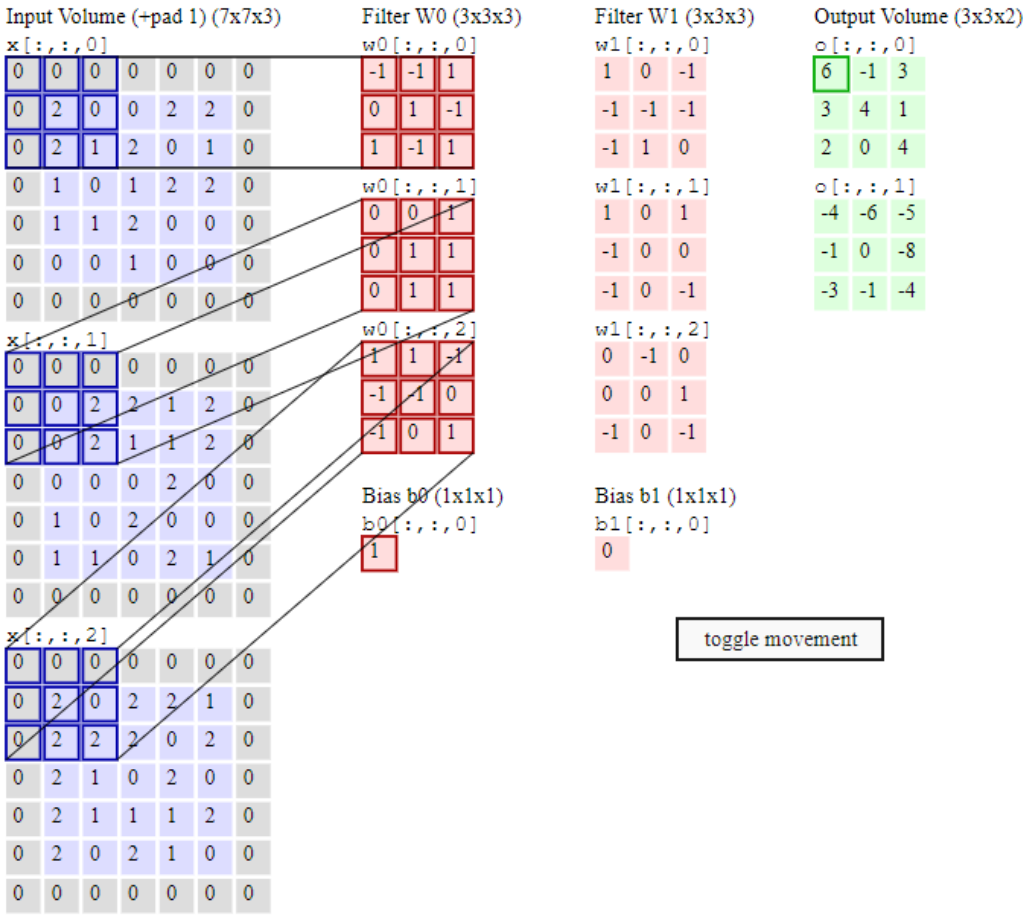
3.3.4 Classificação de imagens

A última etapa do projeto é a classificação das imagens. Para isso foi utilizada uma rede neural convolucional composta por cinco grupos de camadas, mas antes das imagens das regiões de interesse passarem pelo classificador será feito um redimensionamento para gerar imagens de 230x230 pixels e conversão de cores para a escala de cinza. O primeiro grupo no qual a imagem a ser analisada passou consiste em 128 filtros convolucionais de tamanho 3x3 pixels, *stride* (quantidade de pixels que o filtro de convolução se deslocou após cada filtragem) igual a 1 e função de ativação ReLU, que explicando de forma simplificada converte valores negativos em 0 e mantém os positivos. Após a função de ativação, a saída do filtro gerou 128 imagens de tamanho 228x228 pixels e então é aplicado um *max pooling* de tamanho 2x2 pixels com a finalidade de diminuir as dimensões das imagens filtradas. O processo descrito acima refere-se ao primeiro grupo de camadas e a quantidade de imagens resultantes é igual ao número de filtros usados na última filtragem. O processo de convolução pode ser visto no exemplo da Figura 10.

A composição do grupo de camadas 2 é semelhante ao grupo 1, este também foi constituído por uma camada de convolução com função de ativação ReLU e um filtro *max pooling* 2x2 no final, porém para os grupos de camadas 3 e 4 o número de filtros na camada de convolução foi de 64 filtros e no grupo de camadas 5 foi de 32 filtros. No final de todo o processo de convolução, foi realizado o procedimento de

flatten que gera um vetor unidimensional com os valores de cada pixel do agrupamento das imagens resultantes e o vetor passa por uma rede neural densa de 512 nós na primeira camada, 512 nós na segunda camada e 2 nós na última camada. As camadas 1 e 2 da rede neural densa possuem função de ativação ReLU e a última *Softmax*. A equação (3) é a função de perda que a rede utilizou para a atualização dos pesos na etapa de treino do modelo, denominada *Categorical Cross-Entropy*.

Figura 10 - Exemplo do processo de convolução.



Fonte: (CS231n Convolutional Neural Networks for Visual Recognition n.d.)

3.4 Arquiteturas de Redes Neurais Convolucionais

Dentro das redes convolucionais temos variadas estruturas que visam solucionar problemas diferentes a partir de variáveis diferentes como por exemplo: Quantidade de Camadas, tamanho do filtro, *Stride*, *Learning Rate* etc. Nesta secção temos como foco mencionar diferentes estruturas e suas vantagens.

3.4.1 VGG FACE

Este foi o primeiro modelo de uma rede convolucional com a finalidade específica de identificação de faces, e para a realização de tal foi feito inicialmente um banco apenas com informações de atores de políticos famosos, onde as imagens destes passaram por processos rigorosos de avaliação e filtragem com o intuito de gerar um banco de dados limpo e eficiente. Na Tabela 5 é possível ver o processo em que os dados foram expostos, e no final a quantidade de imagens que será utilizada no treinamento (Parkhi, Vedaldi e Zisserman 2015).

Tabela 5 – *Datasets* após cada estágio de processamento.

Stage	Aim	Type	# of persons	# of images per person	total # images	Annotation effort	100%-EER
1	Candidate list generation	A	5,000	200	1,000,000	–	–
2	Image set expansion	M	2,622	2,000	5,244,000	4 days	–
3	Rank image sets	A	2,622	1,000	2,622,000	–	96.90
4	Near dup. removal	A	2,622	623	1,635,159	–	–
5	Final manual filtering	M	2,622	375	982,803	10 days	92.83

Fonte: (Parkhi, Vedaldi e Zisserman 2015)

No desenvolvimento do treinamento para este *dataset* foram projetadas 3 arquiteturas (A, B, D). A Arquitetura “A” é sempre composta por 11 camadas (8 destas são convolucionais e as últimas 3 são redes neurais densas), compostas estrategicamente por funções ReLU e algumas camadas *Max Pooling*. As Arquiteturas “B” e “D” são semelhantes a arquitetura A, porém estás possuem de 2 a 5 camadas adicionais dentre as camadas convolucionais. A entrada para todas as redes é uma imagem de rosto de tamanho 224 x 224 com imagem de rosto média (calculada a partir do conjunto de treinamento) subtraída - isso é crítico para a estabilidade do algoritmo de otimização (Parkhi, Vedaldi e Zisserman 2015).

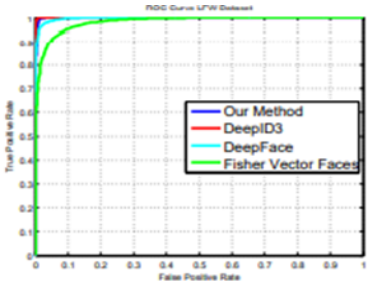
Tabela 6 - Arquiteturas e suas respectivas acurácias.

No.	Config	Data	Train Align.	Test Align.	Embedding	100% - EER
1	A	C	No	No	No	92.83
2	A	F	No	No	No	95.80
3	A	F	No	Yes	No	96.70
4	B	F	No	Yes	No	97.27
5	B	F	Yes	Yes	No	96.17
6	D	F	No	Yes	No	96.73
7	B	F	No	Yes	Yes	99.13

Fonte: (Parkhi, Vedaldi e Zisserman 2015)

Figura 11 - Comparação com o Estado-da-Arte.

No.	Method	Images	Networks	Acc.
1	Fisher Vector Faces [21]	-	-	93.10
2	DeepFace [29]	4M	3	97.35
3	Fusion [30]	500M	5	98.37
4	DeepID-2,3		200	99.47
5	FaceNet [22]	200M	1	98.87
6	FaceNet [22] + Alignment	200M	1	99.63
7	Ours	2.6M	1	98.95



Fonte: (Parkhi, Vedaldi e Zisserman 2015)

Pode-se observar que este projeto alcançou resultados comparáveis ao estado de a arte usando muito menos dados e uma arquitetura de rede muito mais simples (Parkhi, Vedaldi e Zisserman 2015).

Capaz de montar um conjunto de dados em grande escala, com pequeno ruído de rótulo, minimizando a quantidade de anotação manual envolvida. Um dos focos principais era usar classificadores mais fracos para classificar os dados apresentados aos anotadores. Este procedimento foi desenvolvido para faces, mas é evidentemente adequado para outras classes de objetos, bem como para tarefas de granulação fina. A segunda contribuição era mostrar que uma CNN profunda, sem enfeites, mas com formação adequada, pode obter resultados comparáveis ao estado da arte.

3.4.2 VGG FACE 2

A Rede VGG FACE 2 é semelhante a VGG FACE, porém o que muda nesta é que é considerado para cada indivíduo aspectos como poses, idade e etnia. Através desta nova estratégia busca-se concluir uma rede com uma eficácia ainda maior do

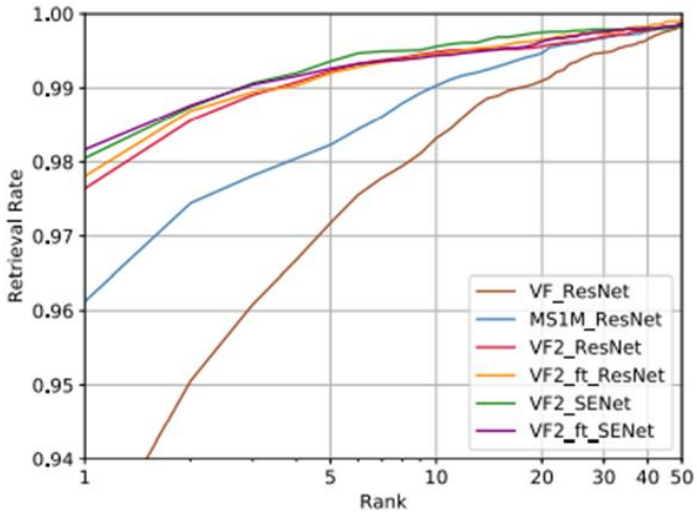
que a primeira. O *Dataset* deste trabalho é composto por imagens de aproximadamente 9000 pessoas, o número total de imagens é algo em torno de 3 Milhões, estas imagens são expostas também a filtros cuja finalidade é diminuir *the "Label Noise",* após tudo isso é possível conferir nos resultados que VGG2 superou VGG1 (Cao, et al. 2017).

Tabela 7 – Comparativo entre os *Datasets*.

Datasets	# of subjects	# of images	# of images per subject	manual identity labelling	pose	age	year
LFW [10]	5,749	13,233	1/2.3/530	-	-	-	2007
YTF [24]	1,595	3,425 videos	-	-	-	-	2011
CelebFaces+ [21]	10,177	202,599	19.9	-	-	-	2014
CASIA-WebFace [26]	10,575	494,414	2/46.8/804	-	-	-	2014
IJB-A [13]	500	5,712 images, 2,085 videos	11.4	-	-	-	2015
IJB-B [23]	1,845	11,754 images, 7,011 videos	36.2	-	-	-	2017
IJB-C [14]	3,531	31,334 images, 11,779 videos	36.3	-	-	-	2018
VGGFace [17]	2,622	2.6 M	1,000/1,000/1,000	-	-	Yes	2015
MegaFace [12]	690,572	4.7 M	3/7/2469	-	-	-	2016
MS-Celeb-1M [7]	100,000	10 M	100	-	-	-	2016
UMDFaces [5]	8,501	367,920	43.3	Yes	Yes	Yes	2016
UMDFaces-Videos [4]	3,107	22,075 videos	-	-	-	-	2017
VGGFace2 (**this paper**)	9,131	3.31 M	80/362.6/843	Yes	Yes	Yes	2018

Fonte: (Cao, et al. 2017)

Figura 12 - Gráfico de performance das redes



Fonte: (Cao, et al. 2017)

No gráfico acima é possível concluir que VGG2 apresenta um desempenho superior a VGG1.

3.4.3 LENET5:

A Lenet funciona com 7 camadas, onde todas possuem parâmetros treináveis. A entrada é uma imagem 32x32 *pixels*, o que é maior que as imagens do *dataset* 20x20. Os valores de entrada são normalizados, com isso o valor do *background level* (*White*) corresponde a um valor de -0,1 e o *foreground* (preto) corresponde a 1,175; isto gera uma média de entrada de aproximadamente 0 e uma variância de aproximadamente 1, o que acelera o aprendizado. Dentre as camadas presentes, a rede C1 é uma rede convolucional com 6 *feature maps* (todas conectadas com 5x5 *neighborhood* na entrada), o tamanho dos *feature maps* são 28x28, o que previne conexões da entrada de cair da fronteira. C1 apresenta 156 parâmetros e 122.304 conexões treináveis. S2 é uma *sub-sampling* com 6 *feature maps* com tamanho de 14x14, estes por sua vez estão conectados a um vizinho com dimensões 2x2 que corresponde a um *feature map* em C1. Quatro entradas são conectadas a um S2, depois multiplicadas por um coeficiente treinável e por fim adicionadas a um bias treinável. O resultado disto é adicionado a uma função sigmoidal, S2 possui 12 parâmetros treináveis e 5.880 conexões. *Layer* C3 é uma rede convolucional com 16 *feature maps*. Cada *feature map* é conectado a um vizinho com dimensões (5x5) com localidades idênticas ao *subset* S2's. Apresenta 1.516 parâmetros treináveis e 151.600 conexões. *Layer* S4 é um *sub-sampling layer* com 16 *feature maps* com tamanho de 5x5. Cada unidade de *feature map* está conectada a um vizinho com dimensões 2x2 no correspondente *feature m*ap em C3. S4 possui 23 parâmetros treináveis e 2.000 conexões. *Layer* C5 Rede convolucional com um 120 *feature maps*. Cada unidade esta conectada a um vizinho com dimensões 5x5 em todos 16 dos *feature maps* de S4. C5 possui 48.120 conexões treináveis. *Layer* F6 contém 84 unidades e é conectado com todas as conexões de C5. Este possui 10.164 parâmetros treináveis (Lecun, et al. 1998).

Como uma Rede Neural, os *Layers* no F6 irão computar um produto entre o vetor de entrada e seus pesos de vetores onde um bias é adicionado. Esta soma, denominada "Ai", é passada por uma função sigmoidal para produzir o estado da unidade "i", chamado de xi (Lecun, et al. 1998).

$$xi = f(ai) \quad (8)$$

A *squashing function* é escalada como tangente hiperbólica:

$$F(a) = Atanh(Sa) \quad (9)$$

A camada de saída é composta através da Função de Base Radial Euclidiana, uma para cada classe, com 84 entradas cada. A saída de cada unidade da função “yi” é computada como:

$$yi = \sum_j (xj - wij) \quad (10)$$

A saída de RBF pode ser interpretada como a probabilidade logarítmica negativa não normalizada de uma distribuição gaussiana no espaço de configurações, da camada F6 (Lecun, et al. 1998).

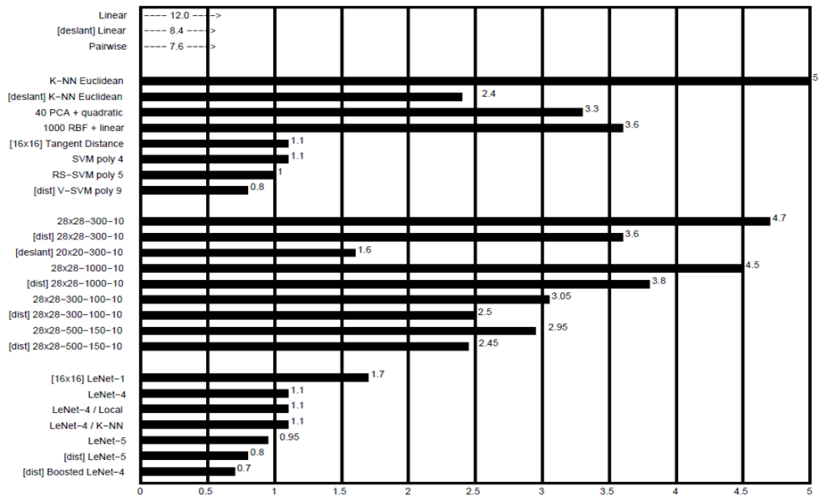
3.4.4 *DataSet*: *The modified NIST set*

Para o treinamento do *Dataset* da rede LeNet5 foi utilizado o NIST modificado, este que é um banco de dados que abrange uma quantidade alta de dígitos. O NIST é constituído por um *Special Database* 3 e um *Special Database* 1, contendo imagens binárias de dígitos *randwritten*. O NIST SD3 (*Special Database* 3) foi designado para o seu conjunto de treinamento e o NIST SD1 (*Special Database*1) para o seu conjunto de teste. Como o *Special Database* 3 é mais limpo e fácil de se reconhecer que o *Special Database*1, um novo *dataset* foi construído a partir dos dois, já que para se tirar conclusões sensatas de experimentos de aprendizagem é preciso que a escolha do conjunto de treinamento e teste não afete os resultados. O banco de dados resultante foi chamado de MNIST. O tamanho das imagens originais em preto e branco foram normalizadas para caber em uma caixa de 20x20 *pixels*, preservando sua proporção de aspecto (Lecun, et al. 1998).

A figura abaixo mostra um a taxa de erro bruta dos classificadores no conjunto de teste de 10.000 exemplos. O *Boosted* LeNet4 apresentou o melhor desempenho

com uma margem de erro de 0,7%, seguindo de perto pelo LeNet-5 com 0,8% (Lecun, et al. 1998).

Figura 13 - Taxa de erro das redes



Fonte: (Lecun, et al. 1998)

4. Resultados

4.1 PROCEDIMENTOS

A primeira etapa do desenvolvimento prático do projeto foi a criação de diversos bancos de dados a partir do banco de imagens dos neonatos da UNIFESP. O banco de imagens possuía originalmente 238 imagens de recém-nascidos com dor e 122 imagens sem dor. Foi desenvolvido um programa na linguagem Python que detectava e recortava a área da face dos neonatos utilizando o algoritmo de detecção facial *RetinaFace.* Após a extração, as imagens das faces foram segmentadas em três outras imagens, para as regiões do olho esquerdo, olho direito e boca. Em seguida, cada segmento foi redimensionado para proporções de 32x32 *pixels*, sendo que para imagens retangulares foi usado preenchimento com *pixels* de valor igual 0 antes do redimensionamento, com intuito de transformá-las em imagens quadradas e evitar distorções. Após a uniformização das dimensões, foi aplicado o processo de *data augmentation,* onde cada imagem foi rotacionada de forma aleatória entre -50 e 50 graus, gerando uma imagem sem rotação e duas imagens rotacionadas. O processo de *data augmentation* gerou 714 imagens com dor e 366 imagens sem dor para olho esquerdo, olho direito e boca.

Com as imagens resultantes, foi criado um *dataset* para o olho esquerdo, olho direito e boca. Para cada um deles foram reservadas 20% das imagens para realização do teste da rede neural e 80% para o treinamento, sendo que cada imagem recebe um rótulo indicando imagem “com dor” ou “sem dor”. A seguir é apresentado uma tabela com as etapas da construção dos *datasets:*

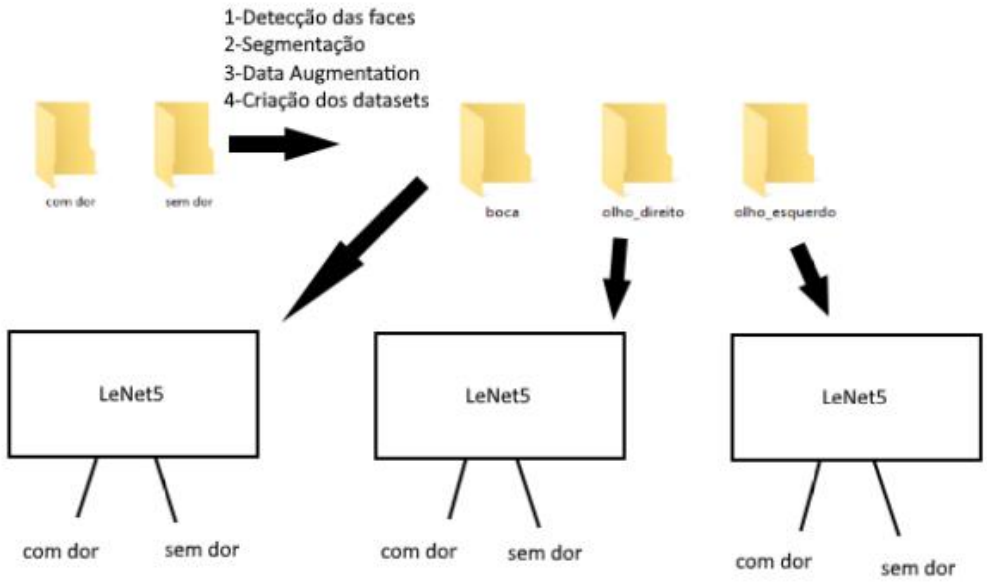
Tabela 8 – Etapas de construção dos *datasets.*

ETAPA	QUANTIDADE DE IMAGENS INICIAL	QUANTIDADES DE IMAGENS RESULTANTE
Detecção da face e segmentação das 3 regiões de interesse	238 imagens com dor 122 imagens sem dor	238 imagens com dor para olho direito, olho esquerdo e boca. 122 imagens sem dor para olho direito, olho esquerdo e boca.
Data augmentation	238 imagens com dor para olho direito, olho esquerdo e boca. 122 imagens sem dor para olho direito, olho esquerdo e boca.	Cada um dos 3 *datasets* possui: 714 imagens com dor 366 imagens sem dor
Separação de 20% das imagens para teste	Cada um dos 3 *datasets* possui: 714 imagens com dor 366 imagens sem dor	Teste: 142 imagens com dor e 73 imagens sem dor Treinamento: 572 imagens com dor e 293 imagens sem dor

Fonte: Autor

Após o processo de criação dos 3 *datasets* foi realizado o treinamento das três redes neurais. A arquitetura das redes utilizadas foi a Lenet5, que é constituída por camadas convolucionais com *kernels* de dimensões 5x5, *stride* igual a 1 e possuem função de ativação do tipo tangente hiperbólica; camadas de *pooling* do tipo *average* com dimensões 2x2 pixels e *stride* igual a 2 e por último 3 camadas de redes neurais densas com funções de ativação do tipo tangente hiperbólica, com respectivamente 120, 84 e 2 neurônios. Esta arquitetura foi escolhida pois foi uma das primeiras redes neurais convolucionais construídas e as imagens usadas em seu treinamento possuíam dimensões de 28x28 pixels, valores próximos às imagens utilizadas neste projeto.

Figura 14 - Representação da criação dos *datasets.*



Fonte: Autor

A primeira camada convolucional recebeu imagens com dimensões 32x32 *pixels* e 3 canais, gerando imagens com dimensões 28x28 *pixels* e 6 canais. A dimensão das imagens geradas pode ser calculada usando a fórmula ilustrada na figura abaixo:

Figura 15 - Equação da dimensão das imagens geradas.

$$n_{out} = \left\lfloor \frac{n_{in} + 2p - k}{s} \right\rfloor + 1$$

n_{in}: number of input features
n_{out}: number of output features
k: convolution kernel size
p: convolution padding size
s: convolution stride size

Fonte: (Balakrishnakumar 2020)

Foi aplicado um filtro do tipo *average pooling* de 2x2 *pixels* e *stride* igual a 2 nas imagens geradas pela primeira camada, transformando-as em imagens com dimensões 14x14 *pixels*, mantendo 6 canais. Depois de passarem pelas 2 camadas convolucionais e *poolings* restantes, foram criadas imagens com 1x1 *pixels* e 120 canais, que posteriormente foram inseridas na rede densa para a classificação.

4.2 RESULTADOS

O primeiro treinamento das redes foi feito utilizando valor de *batch size* igual a 1, *learning rate* igual a 0.01, momentum igual a 0 e foram utilizados 150 *epochs.* Em seguida as redes foram testadas, obtendo-se valores de acerto apresentados na tabela a seguir:

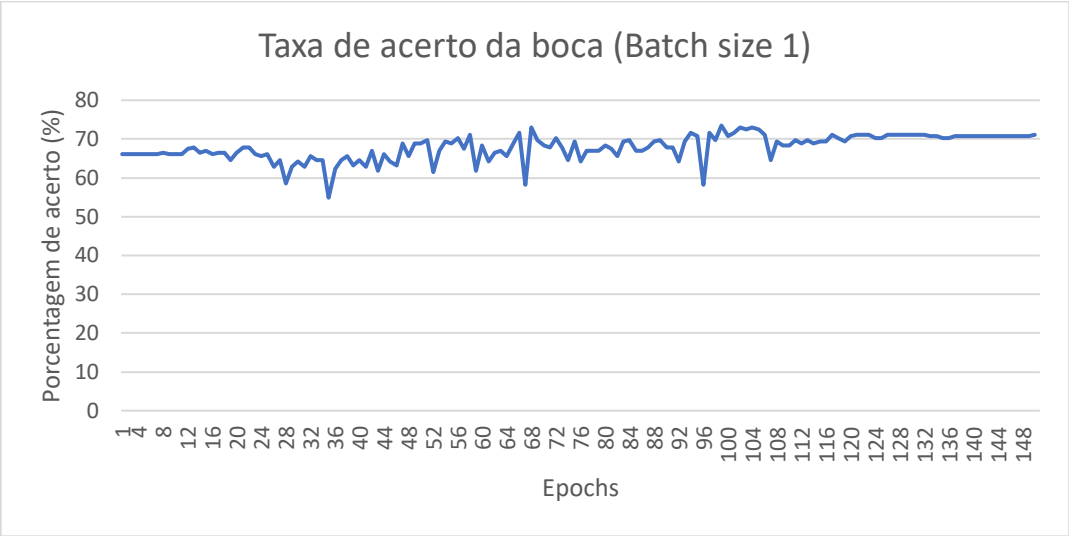
Tabela 9 - Valores dos parâmetros utilizados para o primeiro treino.

Batch size	1
Learning rate	0.01
Momentum	0
Acerto da Boca	71,16 %
Acerto do Olho Esquerdo	76,74 %
Acerto do Olho Direito	76,74 %

Fonte: Autor

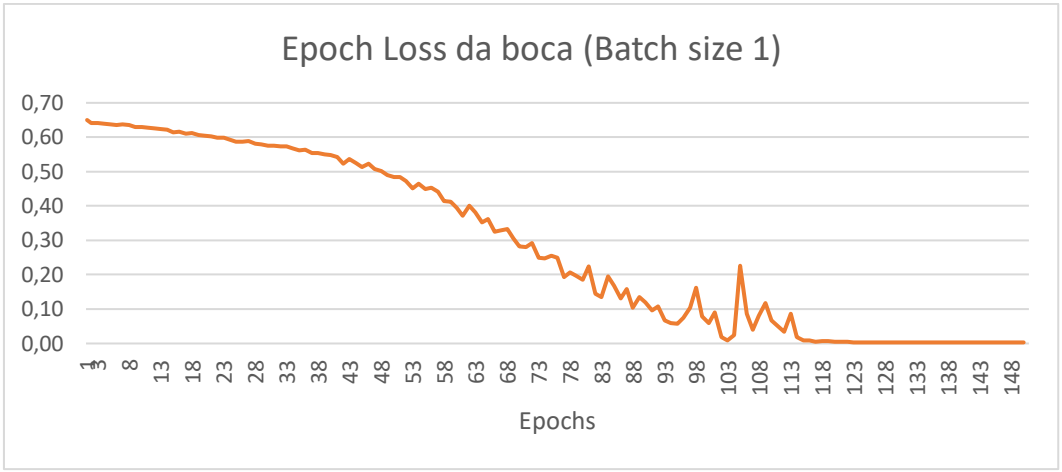
A seguir serão apresentados os gráficos da porcentagem de acerto e do erro em função dos *epochs* para as redes da boca, olho esquerdo e olho direito.

Figura 16 - Gráfico da taxa de acerto das imagens da boca (*Batch size* 1)



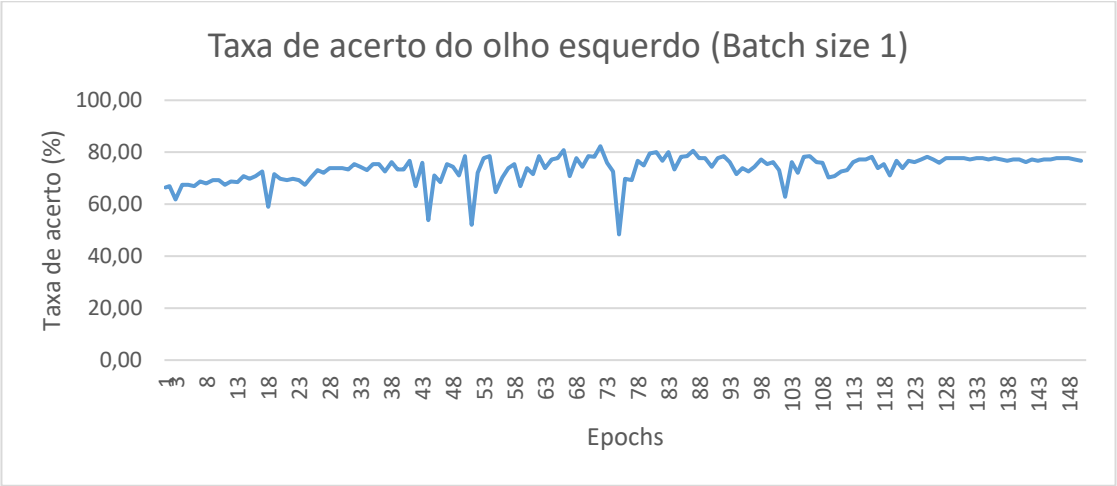
Fonte: Autor

Figura 17 - Gráfico de erro para a boca (*Batch size* 1)



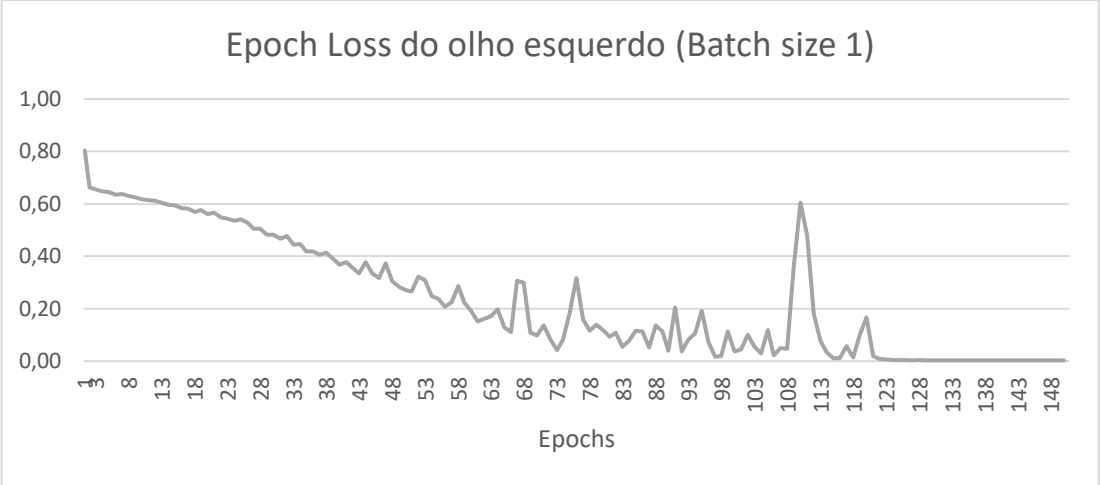
Fonte: Autor

Figura 18 - Gráfico da taxa de acerto das imagens do olho esquerdo (*Batch size* 1)



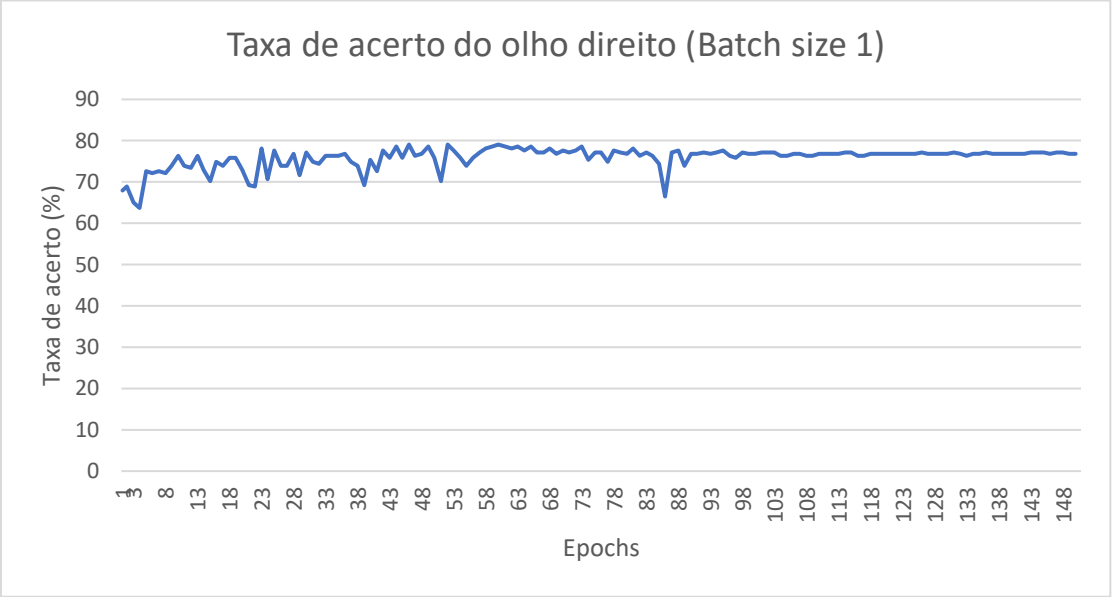
Fonte: Autor

Figura 19 - Gráfico de erro para o olho esquerdo (*Batch size* 1)



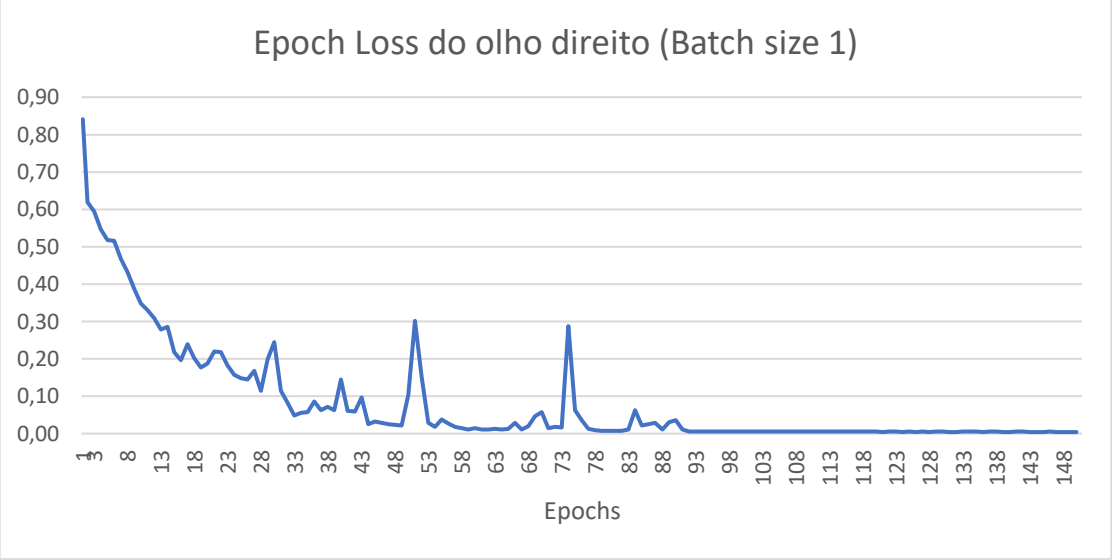
Fonte: Autor

Figura 20 - Gráfico da taxa de acerto das imagens do olho direito (*Batch size* 1)



Fonte: Autor

Figura 21 - Gráfico de erro para o olho direito (*Batch size* 1)



Fonte: Autor

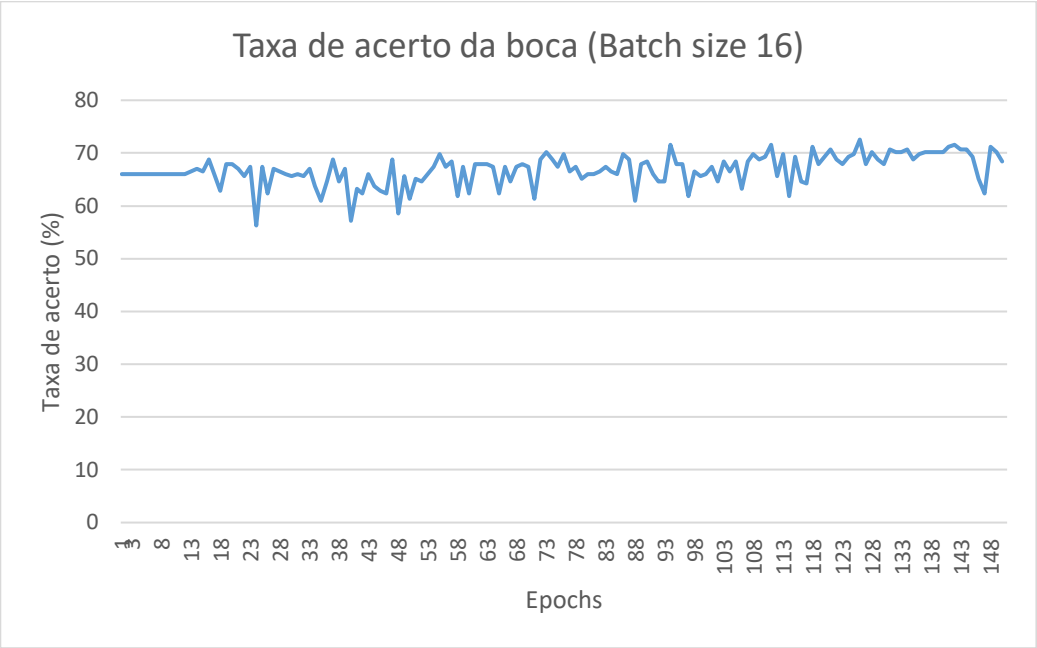
Depois da realização dos testes para as condições citadas anteriormente, o *batch size* foi alterado para 16 e o momentum do otimizador alterado para 0,9; mantendo os valores do *learning rate* e o número de *epochs* do treinamento anterior. Os resultados obtidos do segundo teste serão apresentados a seguir na Tabela 10 e nos gráficos de erro e acerto de cada rede neural treinada.

Tabela 10 - Valores dos parâmetros utilizados para o segundo treino.

Batch size	16
Learning rate	0.01
Momentum	0.9
Acerto da Boca	68,37 %
Acerto do Olho Esquerdo	68,84 %
Acerto do Olho Direito	65,58 %

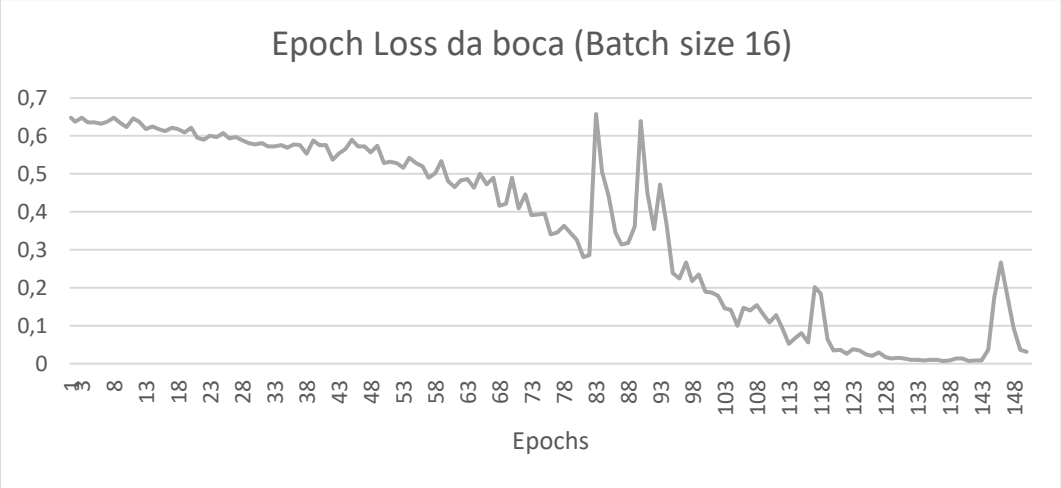
Fonte: Autor

Figura 22 - Gráfico da taxa de acerto das imagens da boca (*Batch size* 16)



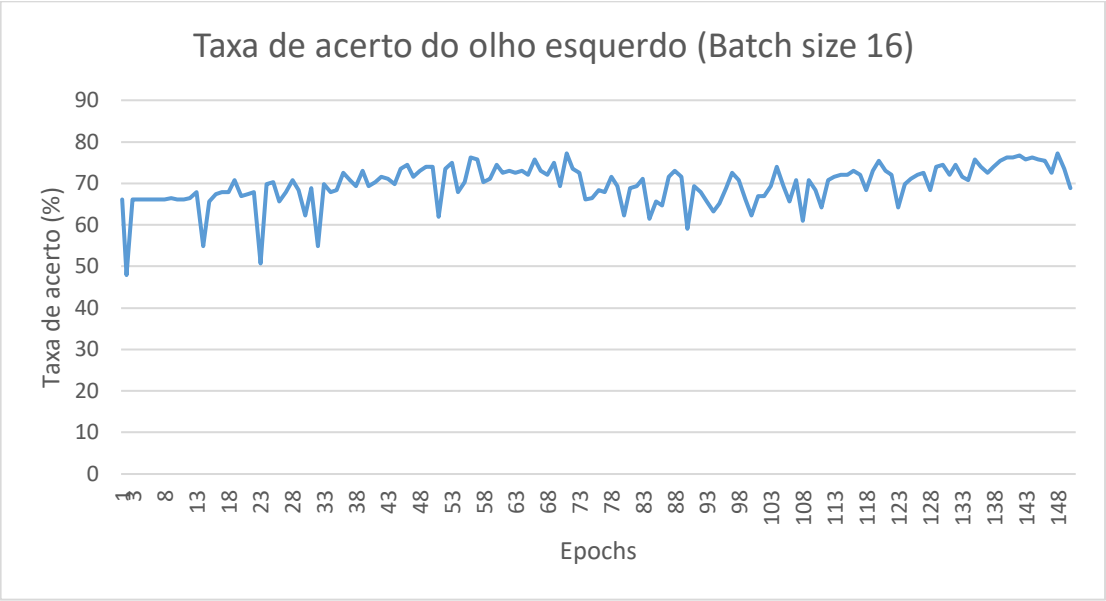
Fonte: Autor

Figura 23 - Gráfico de erro para a boca (*Batch size* 16)



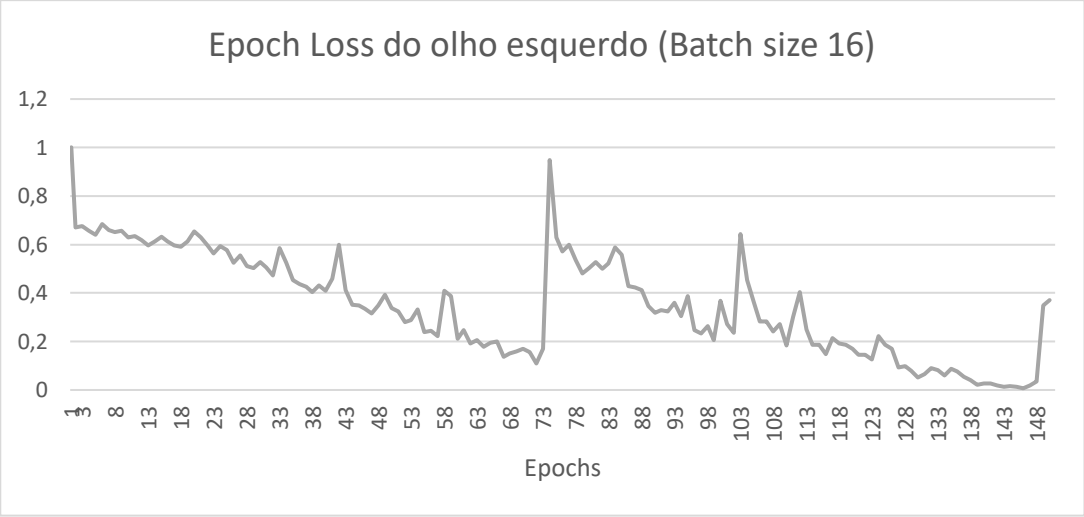
Fonte: Autor

Figura 24 - Gráfico da taxa de acerto das imagens do olho esquerdo (*Batch size* 16)



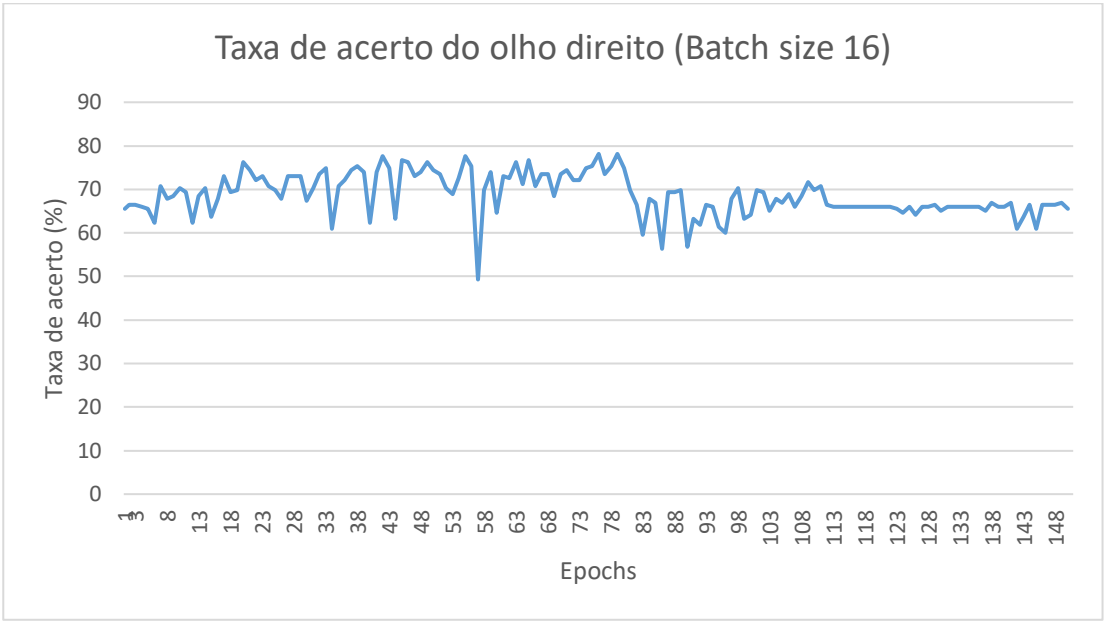
Fonte: Autor

Figura 25 - Gráfico de erro para o olho esquerdo (*Batch size* 16)



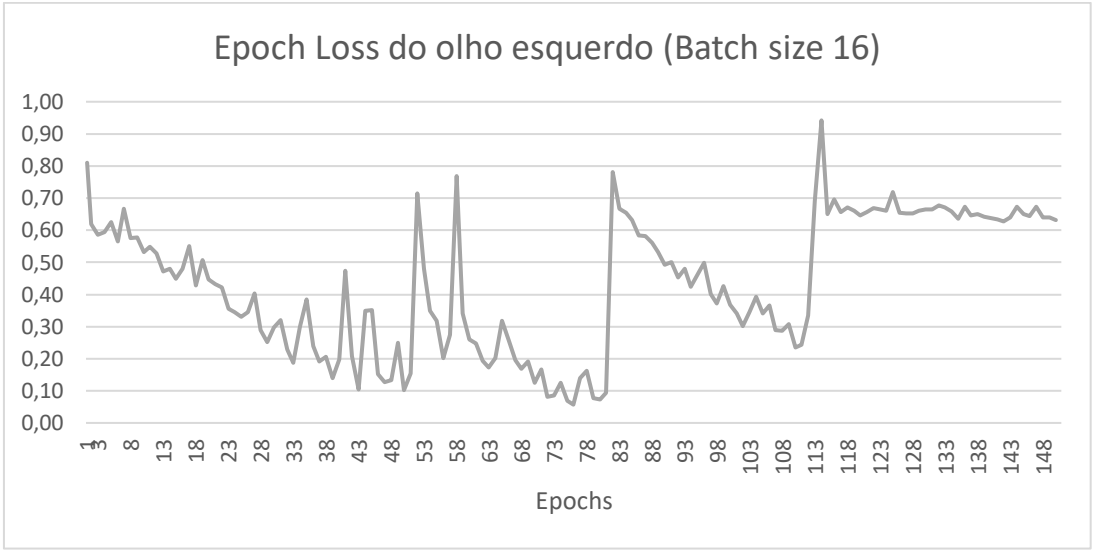
Fonte: Autor

Figura 26 - Gráfico da taxa de acerto das imagens do olho direito (*Batch size* 16)



Fonte: Autor

Figura 27 - Gráfico de erro para o olho direito (*Batch size* 16)



Fonte: Autor

Ao observar os resultados das duas etapas de testes, percebemos que o segundo teste obteve porcentagens de acerto menores que o primeiro teste, isso devido ao *batch,* pois ao aumentá-lo, diminui a acurácia do teste. Podemos considerar os resultados do teste 1 próximos a 70% como satisfatórios pois a arquitetura de rede neural utilizada (LeNet5) foi inicialmente construída para reconhecimento de dígitos manuscritos utilizando a base de dados MNIST, que é constituída por 70 mil imagens com dimensões 28 por 28 *pixels* e 1 camada. Pensando na aplicabilidade do projeto em situação real, podemos criar diversos modelos para identificação da dor nos recém-nascidos. Uma ideia de implementação seria fazer uma lógica booleana do tipo "OU" com as saídas das redes, usando o valor lógico 1 para situação "com dor" e 0 para "sem dor". Desse modo, se a rede do olho esquerdo, olho direito ou boca indicar situação de dor, conclui-se que o neonato está sentindo dor. Outras ideias de implementação seriam a mensuração da quantidade de redes indicando situação com "dor", considerando a classificação da maioria e aplicação da lógica AND para diminuir os falsos positivos.

Devemos também considerar que a rede foi treinada com imagens de recém-nascidos sem aparatos médicos na face e isto pode aumentar as hipóteses de diminuição na porcentagem de acertos caso o neonato possua um artefato em situação real. Algumas propostas de melhorias de performance da rede seriam a utilização de outra arquitetura de rede neural convolucional, como a VGG Face que foi pré-treinada e possui mais camadas de convoluções do que a LeNet5. O aumento do *dataset* utilizando outras transformações de imagem no *data augmentation* e a inclusão de imagens de recém nascidos com artefatos também podem gerar melhores resultados.

5. Conclusão

Este trabalho abordou e implementou redes neurais para a detecção de dor, por meio da análise das regiões faciais neonatais.

O experimento, ao trabalhar com uma base de dados reduzidas da UNIFESP, demonstrou resultados satisfatórios que validam a eficácia da identificação da presença de dor nos recém-nascidos, levando em consideração que foram utilizados os melhores parâmetros para treinamento. Os resultados obtidos apresentam uma acurácia média de 71,16% para a boca e 76,74% para ambos os olhos, variando conforme os parâmetros aplicados no teste. Além disso, a ferramenta demonstrou a capacidade de distinguir e segmentar as faces a partir de um banco de imagens.

Como trabalhos subsequentes, vislumbra-se a ampliação do banco de dados de forma a incluir imagens de neonatos com maiores variações, tais como a presença de artefatos em suas faces. Além disso, recomenda-se a adição de mais regiões do rosto que são capazes de diagnosticar a expressão de dor, proporcionando mais dados para análise e consequentemente uma maior acurácia nos resultados.

Referência Bibliográfica

ANAN D, K.J .S, e K.D C RAIG. 1996. “New perspectivies on the definition of the pain.” Em *Pain*, por K.J .S ANAN D e K.D C RAIG, 3-6.

2018. *Portal de Boas Práticas em Saúde da Mulher, da Criança e do Adolescente.* 11 de Dezembro. https://portaldeboaspraticas.iff.fiocruz.br/atencao-recem-nascido/principais-questoes-dor-em-rn/.

Hall, Richard W, e Kanwaljeet JS Anand. 2015. *US National Library of Medicine.* Acesso em 15 de Abril de 2021. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4254489/.

Werner, Philipp, AYOUB AL-HAMADI, e ROBERT NIESE. 2014. “COMPARATIVE LEARNING APPLIED TO INTENSITY RATING OF FACIAL EXPRESSIONS OF PAIN.” *International Journal of Pattern Recognition and Artificial Intelligence* 28 (5).

Zamzmi, Ghada, Rahul Paul, Dmitry Goldgof, Rangachar Kasturi, e Yu Sun. 2019. “Pain Assessment From Facial Expression: Neonatal Convolutional Neural Network (N-CNN).” *2019 International Joint Conference on Neural Networks (IJCNN)*, 1-7.

HEIDERICH, Tatiany Marcondes, Ana Teresa Figueiredo Stochero LESLIE, e Ruth GUINSBURG. 2015. “Neonatal procedural pain can be assessed by computer software that has good sensitivity and specificity to detect facial movements.” *Wiley Online Library*, 63-69.

DENG et al, Jiankang. 2020. “RetinaFace: Single-Shot Multi-Level Face Localisation in the Wild.” *PROCEEDINGS of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* 5203-5212.

2021. *CATEGORICAL CROSSENTROPY.* Acesso em 4 de Junho de 2021. https://peltarion.com/knowledge-center/documentation/modeling-view/build-an-ai-model/loss-functions/categorical-crossentropy .

s.d. *CS231n Convolutional Neural Networks for Visual Recognition.* Acesso em 12 de Maio de 2021. https://cs231n.github.io/convolutional-networks/.

Grunau, Ruth V.E., e Kenneth D. Craig. 1986. “Pain expression in neonates: facil action and cry.”

Mitchell, Anita, e Barbara J Boss. 2002. “Adverse effects of pain on the nervous systems of newborns and young children: a review of the literature.” *National Library of Medicine.* Acesso em 20 de Abril de 2021.

Guinsburg, Ruth. 1999. “Avaliação e tratamento da dor no recém-nascido.” *Jornal de Pediatria.*

Alves, Gisely. 2018. *Medium.* Acesso em 15 de Abril de 2021. https://medium.com/neuronio-br/entendendo-redes-convolucionais-cnns-d10359f21184.

Lecun, Y, L Bottou, Y Bengio, e P Haffner. 1998. “Gradient-based learning applied to document recognition.” *IEEE.*

Anand, KJS, Phil D, e Hickey PR. 1987. “Pain and its effects in the human neonate and fetus.” *New England Journal of Medicine* 1321-1329.

LS, Frank, Greenberg CS, e Stevens B. 2000. “Pain assessment in infants and children.” *Pediatric Clinics of North America* 487-512.

Nicolau, Carla, Juliana Pigo, Bueno Bueno, e Falcão Mário. 2008. “Avaliação da dor em recém-nascidos prematuros durante a fisioterapia respiratória.” *Revista Brasileira de Saúde Materno Infantil .*

Heiderich, Tatiany, R Guinsburg, e Ana Leslie. 2013. “Desenvolvimento de Software para Identificar a Expressão Facial de Dor do Recém-Nascido.” Universidade Federal de São Paulo (UNIFESP), São Paulo.

Balakrishnakumar. 2020. 1 de Setembro. https://medium.datadriveninvestor.com/architecture-implementation-of-lenet-from-scratch-in-pytorch-709cc38c00a9.

Lawrence, J, D Alcock, P McGrath, J Kay, S B MacMurray, e C Dulberg. 1993. “The development of a tool to assess neonatal pain.” 12 de Setembro: 59-66.

Stevens, Bonnie, Celeste Johnston, Patricia Petryshen, e Anna Taddio. 1996. “Premature Infant Pain Profile: development and initial validation.” 12 de Março: 13-22.

Parkhi, Omkar M., Andrea Vedaldi, e Andrew Zisserman. 2015. “Deep Face Recognition.” Setembro.

Cao, Qiong, Li Shen, Weidi Xie, Omkar M. Parkhi, e Andrew Zisserman. 2017. “VGGFace2: A dataset for recognising faces across pose and age.” 23 de Outubro.

Zamzmi, Ghada, Rahul Paul, Dmitry Goldgof, Rangachar Kasturi, e Yu Sun. 2017. “A Review of Automated Pain Assessment in Infants.”