



# **APLICAÇÃO DE REDES NEURAIS CONVOLUCIONAIS PARA IDENTIFICAÇÃO DE INDICATIVOS DE DEMÊNCIA EM TESTE COGNITIVO**

**Trabalho de Conclusão de Curso**

**Engenharia da Computação**

**Elisson Lima Gomes da Silva**  
**Orientador: Prof. Carmelo José Albanez Bastos Filho**



**UNIVERSIDADE  
DE PERNAMBUCO**

**Universidade de Pernambuco  
Escola Politécnica de Pernambuco  
Graduação em Engenharia de Computação**

**ELISSON LIMA GOMES DA SILVA**

**APLICAÇÃO DE REDES NEURAIS  
CONVOLUCIONAIS PARA  
IDENTIFICAÇÃO DE INDICATIVOS DE  
DEMÊNCIA EM TESTE COGNITIVO**

Monografia apresentada como requisito parcial para obtenção do diploma de Bacharel em Engenharia de Computação pela Escola Politécnica de Pernambuco – Universidade de Pernambuco.

**Recife, julho 2020.**

Elisson Lima Gomes da Silva

Aplicação de redes neurais convolucionais para identificação de indicativos de demência em testes cognitivos/ Elisson Lima Gomes da Silva – Recife – PE, Brasil, julho 2020 – 58 p.

Orientador: Carmelo José Albanéz Bastos Filho

Trabalho de Conclusão de Curso – Engenharia da Computação

Escola Politécnica de Pernambuco

Universidade de Pernambuco, julho de 2020

1. Redes Neurais Convolucionais 2. Mini Exame do Estado Mental 3. Teste do desenho do pentágono I. Prof. Carmelo José Albanéz Bastos Filho II. Universidade de Pernambuco III. Escola politécnica, IV. Título

## MONOGRAFIA DE FINAL DE CURSO

### Avaliação Final (para o presidente da banca)\*

No dia 6/8/2020, às 16h00min, reuniu-se para deliberar sobre a defesa da monografia de conclusão de curso do(a) discente **ELISSON LIMA GOMES DA SILVA**, orientado(a) pelo(a) professor(a) **CARMELO JOSE ALBANEZ BASTOS FILHO**, sob título **Aplicação de redes convolucionais para identificação de indicativos de demência em teste cognitivo**, a banca composta pelos professores:

**DIEGO JOSÉ RÁTIVA MILLÁN (PRESIDENTE)**

**CARMELO JOSE ALBANEZ BASTOS FILHO (ORIENTADOR)**

Após a apresentação da monografia e discussão entre os membros da Banca, a mesma foi considerada:

Aprovada       Aprovada com Restrições\*       Reprovada

e foi-lhe atribuída nota: **10** ( **Dez** )

\*(Obrigatório o preenchimento do campo abaixo com comentários para o autor)

O(A) discente terá **10** dias para entrega da versão final da monografia a contar da data deste documento.

AVALIADOR 1: Prof (a) **DIEGO JOSÉ RÁTIVA MILLÁN**

AVALIADOR 2: Prof (a) **CARMELO JOSE ALBANEZ BASTOS FILHO**

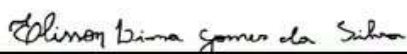
AVALIADOR 3: Prof (a)

\* Este documento deverá ser encadernado juntamente com a monografia em versão final.

## Autorização de publicação de PFC

Eu, **Elisson Lima Gomes da Silva** autor(a) do projeto de final de curso intitulado: **Aplicação de redes convolucionais para identificação de indicativos de demência em teste cognitivo**; autorizo a publicação de seu conteúdo na internet nos portais da Escola Politécnica de Pernambuco e Universidade de Pernambuco.

O conteúdo do projeto de final de curso é de responsabilidade do autor.



\_\_\_\_\_  
**Elisson Lima Gomes da Silva**



\_\_\_\_\_  
Orientador(a): **Carmelo Jose Albanez Bastos Filho**

\_\_\_\_\_  
Coorientador(a):



\_\_\_\_\_  
Prof. de TCC: **Daniel Augusto Ribeiro Chaves**

\_\_\_\_\_  
Data: 6/8/2020

# Resumo

O objetivo deste trabalho é apresentar um estudo da aplicação de diversas arquiteturas de redes neurais convolucionais para automação do teste do desenho do pentágono aplicado no processo de avaliação cognitiva que busca identificar a presença e o estágio da doença de Alzheimer em pessoas idosas. O teste que esse trabalho busca automatizar é aplicado no Mini Exame do Estado Mental (MMSE) que é, atualmente, a avaliação cognitiva mais utilizada para identificar sintomas de demência em pessoas idosas. Uma de suas várias etapas consiste em copiar dois pentágonos sobrepostos onde a interseção desses polígonos forme um losango. Essa figura desenhada pelo paciente será a entrada do modelo convolucional que deverá classificá-la como correto ou incorreto de acordo com os critérios estabelecidos no MMSE. Nos experimentos realizados nesse trabalho foi utilizada uma base de dados coletada com ajuda de voluntários que simularam esta etapa do MMSE e forneceram um bom número de imagens que foram utilizadas para treinamento dos modelos. As melhores arquiteturas testadas obtiveram acurácia média de 93% no conjunto de teste.

**Palavras-chave:** Redes Neurais Convolucionais. Mini Exame do Estado Mental. Teste do desenho do pentágono. Doença de Alzheimer. Aprendizado profundo.

# Abstract

The objective of this work is to present a study of the application of several architectures of convolutional neural networks for automation of the pentagon design test applied in the cognitive assessment process that seeks to identify the presence and stage of Alzheimer's disease in elderly people. The test that this work seeks to automate is applied to the Mini Mental State Examination (MMSE), which is currently the most widely used cognitive assessment to identify dementia symptoms in the elderly. One of its several steps consists of copying two overlapping pentagons where the intersection of these polygons forms a rhombus. This figure drawn by the patient will be the input of the convolutional model that should classify it as correct or incorrect according to the criteria established in the MMSE. In the experiments carried out in this work, a database was collected with the help of volunteers who simulated this stage of the MMSE and provided a good number of images that were used for training the models. The best architectures tested achieved an average accuracy of 93% in the test set.

**Keywords:** Convolutional Neural Networks. Mini-Mental State Examination. Double-diamond copy. Alzheimer's disease. Deep learning.

# Sumário

Índice de Figuras .....	10
Índice de Tabelas .....	13
Tabela de Símbolos e Siglas.....	14
1. Introdução.....	15
1.1. Contextualização .....	15
1.2. Motivação .....	16
1.3. Objetivo Geral.....	17
1.4. Objetivo Específico .....	17
1.5. Organização do Trabalho .....	17
2. Teste cognitivo .....	18
2.1. Mini-Mental State Examination .....	18
2.2. Teste do desenho do pentágono .....	18
2.3. Métodos para automatização do teste do pentágono .....	21
3. Classificação de Imagens e Redes Neurais Convolucionais .....	23
3.1. Usando redes neurais artificiais para classificar imagens .....	23
3.2. Redes neurais convolucionais.....	24
3.2.1. Passo de deslocamento das janelas convolucionais.....	25
3.2.2. <i>Padding</i> .....	26
3.2.3. Camada de <i>Pooling</i> .....	27
4. Experimentos.....	28
4.1. Base de Dados .....	28
4.2. Treinamentos e testes .....	29
4.2.1. Experimentos para validação da arquitetura da rede .....	29
4.2.2. Experimentos com bases normalizadas .....	40



4.2.3. Experimentos com a base normalizada e pré-processada .....	44
4.3. Análise da certeza da rede .....	48
4.4. Resumo dos resultados dos experimentos .....	50
5. Resultados e conclusões .....	53
5.1. Trabalhos futuros .....	55
Bibliografia .....	56

# Índice de Figuras

<b>Figura 1.</b> Pentágonos sobrepostos, teste do MMSE .....	16
<b>Figura 2.</b> Pentágonos desenhados corretamente.....	19
<b>Figura 3.</b> Pentágonos desenhados incorretamente .....	19
<b>Figura 4.</b> Escala de Bourke para interpretação dos resultados do teste do desenho do pentágono. ....	20
<b>Figura 5.</b> Ressonância magnética sujeita a análise de imagem automatizada. O código de cores é usado para indicar as áreas do cérebro que são particularmente informativas para o programa de computador com relação ao diagnóstico diferencial da doença de Alzheimer. (ESCHWEILER, 2010).....	21
<b>Figura 6.</b> Arquitetura de rede neural proposta por LINS (2017) para processo de classificação de pacientes com demência. ....	22
<b>Figura 7.</b> Representação de uma arquitetura básica de uma rede neural convolucional para classificação de dígitos escritos. ....	23
<b>Figura 8.</b> Representação da extração de características da convolução numa rede neural convolucional. (LECUN, 2015). ....	25
<b>Figura 9.</b> Representação da influência do hiperparâmetro stride na operação da convolução. Nessa imagem demonstra o deslocamento da janela convolucional para um valor de stride igual a dois.....	26
<b>Figura 10.</b> Representação da influência do hiperparâmetro stride na operação da convolução. Nessa imagem demonstra o deslocamento da janela convolucional para um valor de stride igual a um. ....	26
<b>Figura 11.</b> Imagem representada pela área branca preenchida com padding igual a 2 (área marrom). ....	27
<b>Figura 12.</b> Representação das operações de Average e Max Pooling.....	27
<b>Figura 13.</b> Imagem obtida com técnicas de data augmentation. ....	28
<b>Figura 14.</b> Imagem obtida com auxílio de voluntários. ....	28

<b>Figura 15.</b> Função sigmoid .....	30
<b>Figura 16.</b> Diagrama da arquitetura aplicada no primeiro experimento do item 4.2.1. .....	31
<b>Figura 17.</b> Gráfico demonstrando desempenho da rede durante treinamento .....	31
<b>Figura 18.</b> Diagrama da arquitetura aplicada no segundo experimento do item 4.2.1. .....	32
<b>Figura 19.</b> Gráfico demonstrando desempenho da rede do segundo experimento durante treinamento .....	33
<b>Figura 20.</b> Diagrama da arquitetura aplicada no terceiro experimento do item 4.2.1. .....	33
<b>Figura 21.</b> Gráfico que demonstra desempenho da rede durante o treinamento do terceiro experimento.....	34
<b>Figura 22.</b> Gráfico que demonstra desempenho da rede no quarto experimento do item 4.2.1.....	35
<b>Figura 23.</b> Diagrama da arquitetura aplicada no quinto experimento do item 4.2.1..	36
<b>Figura 24.</b> Gráfico exibindo desempenho da rede durante o treinamento do quinto experimento.....	37
<b>Figura 25.</b> Diagrama da arquitetura aplicada no sexto experimento do item 4.2.1...	37
<b>Figura 26.</b> Gráfico demonstrando o desempenho da rede do sexto experimento do item 4.2.1.....	38
<b>Figura 27.</b> Gráfico que demonstra desempenho do treinamento da rede no sétimo experimento do item 4.2.1.....	39
<b>Figura 28.</b> Função ReLU. ....	41
<b>Figura 29.</b> Gráfico que demonstra desempenho da rede durante treinamento no primeiro experimento do item 4.2.2.....	41
<b>Figura 30.</b> Gráfico que demonstra desempenho da rede no segundo experimento do item 4.2.2.....	43

<b>Figura 31.</b> Gráfico que demonstra desempenho da rede no terceiro experimento do item 4.2.2.....	44
<b>Figura 32.</b> Gráfico que demonstra desempenho do treinamento da rede no primeiro experimento do item 4.2.3.....	45
<b>Figura 33.</b> Gráfico que demonstra desempenho da rede durante o treinamento do segundo experimento do item 4.2.3. ....	46
<b>Figura 34.</b> Diagrama da arquitetura aplicada no terceiro experimento do item 4.2.3. ....	47
<b>Figura 35.</b> Gráfico que demonstra desempenho da rede durante treinamento do terceiro experimento do item 4.2.3. ....	47
<b>Figura 36.</b> Distribuição da certeza para imagens classificadas corretamente, quanto mais próximo de 1, melhor. Base de dados não pré-processadas.....	49
<b>Figura 37.</b> Distribuição da certeza para imagens classificadas corretamente, quanto mais próximo de 1, melhor. Base de dados pré-processadas.....	49
<b>Figura 38.</b> Imagem incorreta classificada corretamente pelo modelo com erro de 0,0070659. ....	53
<b>Figura 39.</b> Imagem correta classificada corretamente pelo modelo com erro de 0,00123647. ....	54
<b>Figura 40.</b> Exemplo de imagem classificada incorretamente. ....	54

# Índice de Tabelas

<b>Tabela 1.</b> Distribuição das bases de dados .....	28
<b>Tabela 2.</b> Divisão das bases de dados .....	29
<b>Tabela 3.</b> Resumo dos resultados obtidos nos experimentos do item 4.2.1. ....	40
<b>Tabela 4.</b> Resumo dos resultados dos experimentos realizados no item 4.2 .....	52

# Tabela de Símbolos e Siglas

CNN – Convolutional Neural Networks (Redes Neurais Convolucionais)

MLP – Multilayer Perceptron (Perceptron Multicamadas)

MMSE – Mini-Mental State Examination (Mini Exame do Estado Mental)

# 1. Introdução

## 1.1. Contextualização

Em 2050 é estimado que a população mundial acima dos 60 anos será de 2 bilhões. Um dos efeitos negativos do envelhecimento é o aumento do número de pessoas com demência. Embora afete, principalmente, pessoas mais velhas a demência não é uma característica normal do envelhecimento (WHO, 2012).

A síndrome clínica de demência possui sinais e sintomas de ordem neuropsicológica onde o indivíduo pode apresentar déficit de memória, afasia (dificuldade em comunicar-se), apraxia (dificuldade em realizar movimentos), agnosia (dificuldade em identificar pessoas), déficit de atenção e disfunções executivas que como, por exemplo, abstração, flexibilidade cognitiva, inibição, planejamento, organização e adaptação a novos estímulos (GRAND; CASPAR; MACDONALD, 2011).

A demência é uma das maiores causas de invalidez entre as pessoas mais velhas, entretanto em muitos países existe uma falta de conscientização e entendimento resultando em problemas no diagnóstico e até no cuidado dos mais idosos além de impactos econômicos, físicos e psicológicos na sociedade. (WHO, 2012).

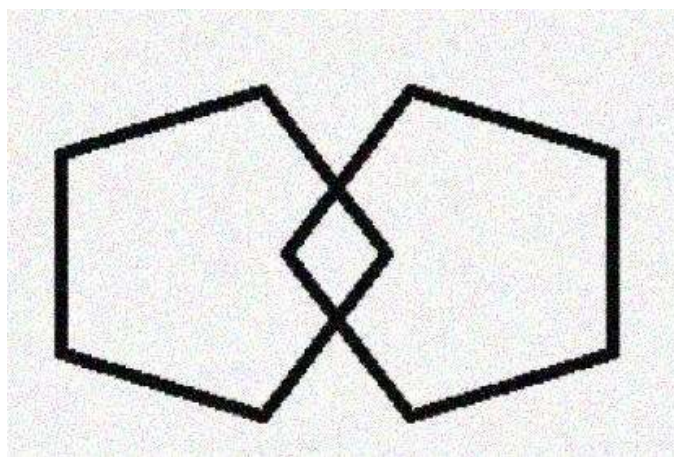
Não existe atualmente tratamento disponível para curar ou até alterar a progressão da demência, existe, no entanto, muito que pode ser oferecido para auxiliar e até melhorar o estilo de vida de pessoas com demência e um dos principais objetivos em seu tratamento é o diagnóstico precoce (WHO, 2012).

## 1.2. Motivação

Identificar pessoas com alto risco para doença de Alzheimer é importante tanto para o paciente quanto para seus familiares e cuidadores, pois isso permite intervenções terapêuticas nos estágios iniciais da doença, o que permite desacelerar ou até prevenir o início do processo da doença (COLLIE; MARUFF, 2002).

Entre os diversos testes cognitivos usados para identificar sintomas de demência em idosos, o mais utilizado no mundo, atualmente, é o *Mini-Mental State Examination* (MMSE) (LINS, 2017). O MMSE foi proposto por Folstein (1975) e é dividido em duas partes, a primeira requer apenas respostas verbais e exige memória, orientação e atenção. A segunda parte testa a habilidade de nomear, seguir instruções verbais e escritas, escrever uma sentença de forma espontânea e copiar um complexo polígono semelhante a figura de *Bender-Gestalt* (FOLSTEIN, 1975).

Este último teste, consiste na tarefa de desenhar dois pentágonos sobrepostos de forma que a sobreposição das figuras geométricas forme um losango (Figura 1), dessa forma a habilidade visual motora do paciente é avaliada (FOUNTOULKIS, 2011).



**Figura 1.** Pentágonos sobrepostos, teste do MMSE



### **1.3. Objetivo Geral**

Este trabalho tem como objetivo utilizar redes neurais convolucionais para classificação de imagens apresentadas durante o MMSE e, com isso, automatizar essa etapa do teste.

### **1.4. Objetivo Específico**

Esse trabalho tem os seguintes objetivos específicos:

- Estudar o funcionamento das Redes Neurais Convolucionais e algumas das arquiteturas mais utilizadas;
- Realizar treinamento de diversas arquiteturas para classificação de conjunto de imagens coletados de voluntários;
- Realizar estudo comparativo e analisar resultados obtidos com as melhores arquiteturas.

### **1.5. Organização do Trabalho**

Esta subseção apresenta a estrutura restante desse trabalho.

O capítulo 2 apresenta uma revisão bibliográfica sobre testes cognitivos e demonstra a utilização do MMSE e do teste do desenho do pentágono.

O capítulo 3 apresenta um breve resumo sobre redes neurais convolucionais, suas principais características e alguns hiperparâmetros.

O capítulo 4 apresenta os diversos experimentos realizados

O capítulo 5 apresenta as conclusões dos resultados e considerações sobre os trabalhos futuros.

## 2. Teste cognitivo

Exames de triagem são bastante utilizados para obter uma breve e objetiva medida da função cognitiva, embora exista um número substancial de exames de triagem o MMSE é o mais utilizado. O *Mini-Mental State* é um exame clínico popular que é disponível em muitos idiomas (TOMBAUGH, 1992).

### 2.1. Mini-Mental State Examination

O MMS é um formulário de exame da situação mental cognitivo, é composto por 11 questões que requer de 5 a 10 minutos para ser aplicado. Ele concentra-se apenas nos aspectos cognitivos das funções mentais e exclui questões sobre humor, experiências mentais anormais e forma de pensar (FOLSTEIN, 1975).

O teste é capaz de refletir mudanças cognitivas clínicas provando ser válido e confiável na identificação de pacientes com demência. O MMSE é útil para estimar quantitativamente a severidade do declínio cognitivo e usado para documentar mudanças cognitivas nos pacientes (FOLSTEIN, 1975).

O resultado final do MMSE é uma pontuação de 0 a 30 que representam a avaliação de 7 categorias que representam diferentes domínios ou funções cognitivas: Orientação ao Tempo (5 pontos), orientação a lugar (5 pontos), Registro de três palavras (3 pontos), atenção e cálculo (5 pontos), memorização de três palavras (3 pontos), linguagem (8 pontos) e construção visual (1 ponto) (TOMBAUGH, 1992).

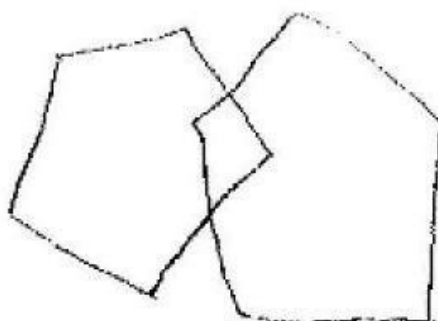
### 2.2. Teste do desenho do pentágono

A habilidade de desenhar e copiar é bastante utilizada para identificação de disfunções cognitivas em idosos, pois é um processo rápido e fácil de ser aplicado, sensível a degenerações iniciais na evolução de doenças e é correlacionada com o funcionamento global cognitivo (CORMAK, 2004).

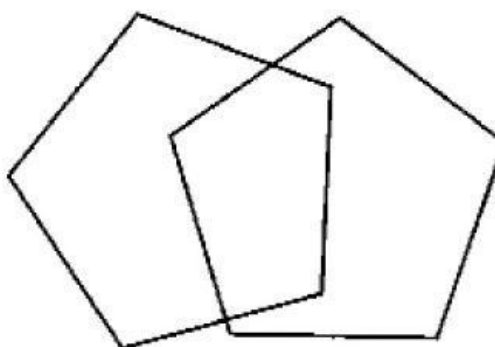
Práxis refere-se à função cognitiva complexa onde o sistema motor é usado para executar ações e movimentos complexos aprendidos como cópia de desenhos. A avaliação da práxis é um importante estágio na avaliação cognitiva porque a apraxia

(dificuldade de realização de movimentos complexos aprendidos) pode estar relacionada a presença de doenças neurodegenerativas como a doença de Alzheimer (MARTINELLI, 2018).

O teste do desenho do pentágono pode ser usado para avaliar se um paciente apresenta ou não apraxia. Existem várias maneiras de interpretar o teste, a primeira forma é a avaliação binária que é usada no MMSE (MARTINELLI, 2018). Nesse método a imagem que apresentar os pentágonos dispostos de forma correta é classificada como correta, conforme Figura 2, e caso contrário a classificação para a imagem é incorreta, conforme Figura 3.



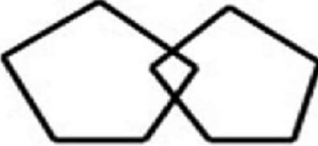


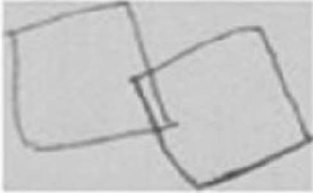

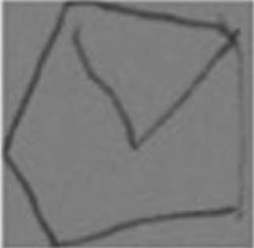

**Figura 2.** Pentágonos desenhados corretamente



**Figura 3.** Pentágonos desenhados incorretamente

A segunda forma de interpretação, proposta por Bourke (1995), atribui uma escala de pontuação para a imagem onde ao desenho correto seria atribuído 6 pontos. Um ponto é dado para linhas desenhadas ou tentativas de desenho de uma figura; 2 pontos para o desenho de uma figura; 3 para duas figuras que não estão sobrepostas;

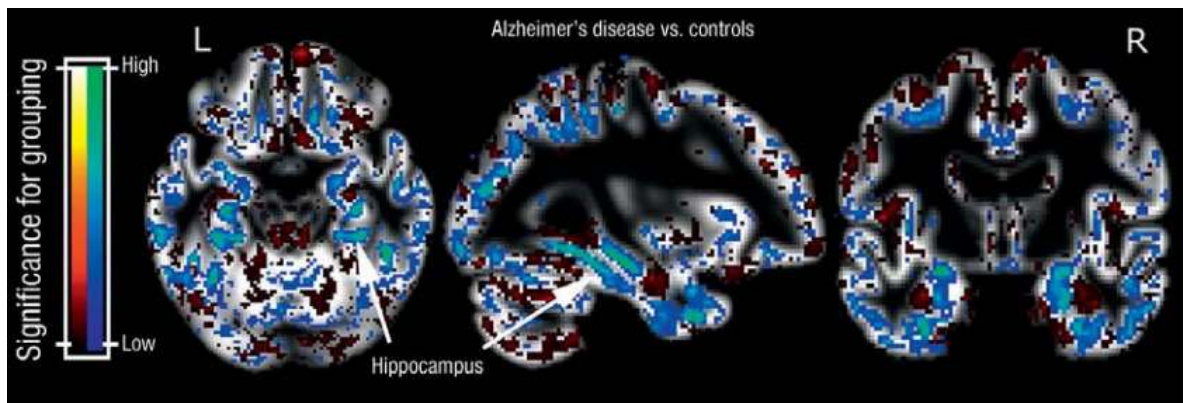
4 pontos para duas figuras sobrepostas; 5 pontos para um pentágono e uma figura sobreposta (MARTINELLI, 2018).

MODELO		MMSE (subteste praxis) = 1 ponto Bourke = 6 pontos
CASO A		MMSE (subteste praxis) = 1 ponto Bourke = 6 pontos
CASO B		MMSE (subteste praxis) = 0 ponto Bourke = 5 pontos
CASO C		MMSE (subteste praxis) = 0 ponto Bourke = 4 pontos
CASO D		MMSE (subteste praxis) = 0 ponto Bourke = 3 pontos
CASO E		MMSE (subteste praxis) = 0 ponto Bourke = 2 pontos
CASO F		MMSE (subteste praxis) = 0 ponto Bourke = 1 pontos

**Figura 4.** Escala de Bourke para interpretação dos resultados do teste do desenho do pentágono.

## 2.3. Métodos para automatização do teste do pentágono

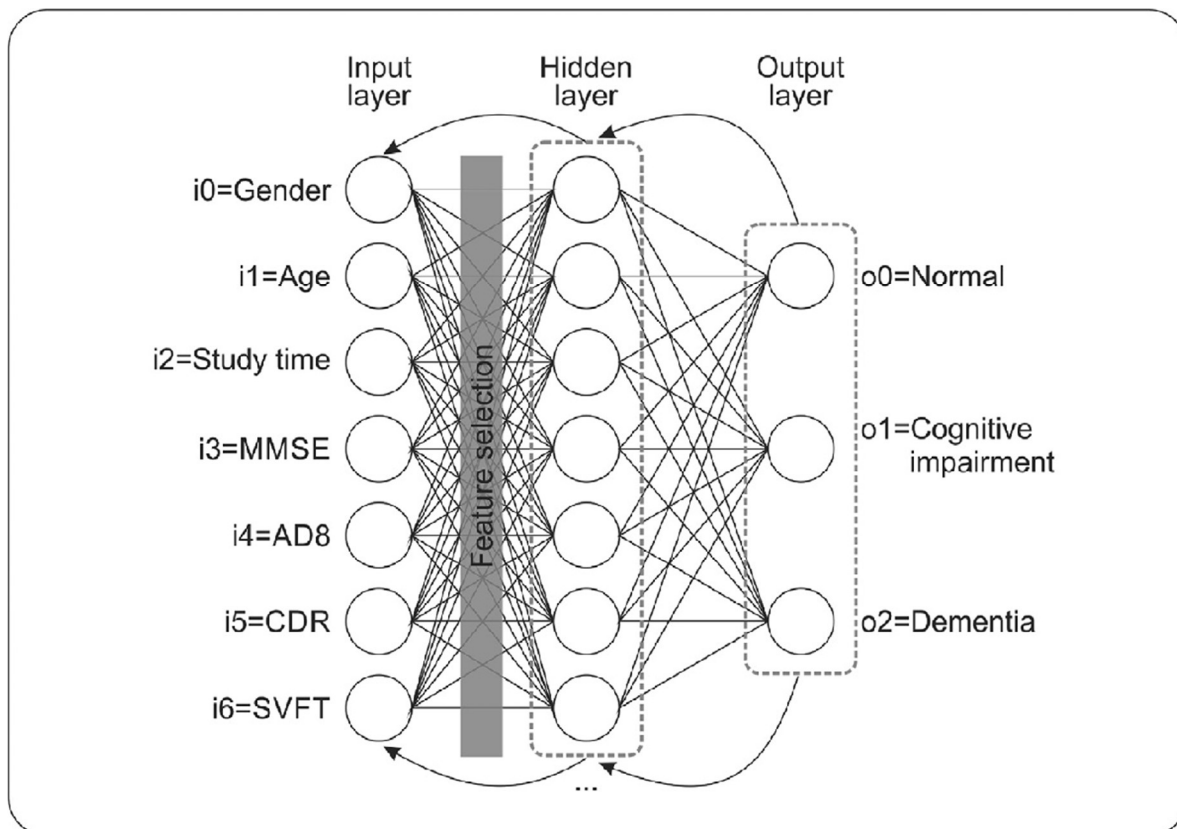
Atualmente existem diversos métodos automáticos, que aplicam aprendizagem de máquina, que auxiliam o diagnóstico de demência, algumas técnicas utilizam máquinas de vetores de suporte que analisam imagens de ressonância magnética e aprendem como distinguir diferentes classes, por exemplo, doença de Alzheimer, demência frontotemporal e envelhecimento normal (Figura 5) (ESCHWEILER, 2010).



**Figura 5.** Ressonância magnética sujeita a análise de imagem automatizada. O código de cores é usado para indicar as áreas do cérebro que são particularmente informativas para o programa de computador com relação ao diagnóstico diferencial da doença de Alzheimer. (ESCHWEILER, 2010).

Outra abordagem utiliza redes neurais artificiais (Figura 6) que recebem informações do paciente, como por exemplo, idade, gênero, tempo de estudo e pontuação no MMSE, e aprendem a distinguir a entrada entre as classes,

envelhecimento normal, declínio cognitivo leve e demência (LINS, 2017).



**Figura 6.** Arquitetura de rede neural proposta por LINS (2017) para processo de classificação de pacientes com demência.

Esse trabalho propõe uma abordagem que automatiza parte do MMSE que é o teste cognitivo mais usado no mundo (TOMBAUGH, 1992), considerado como padrão-ouro (LINS, 2017). Utilizando redes neurais convolucionais (Convolutional Neural Networks - CNN) foi possível automatizar a avaliação do teste do desenho do pentágono aplicado na última etapa do MMSE, dessa forma é possível contribuir para a automatização de todo o MMSE.

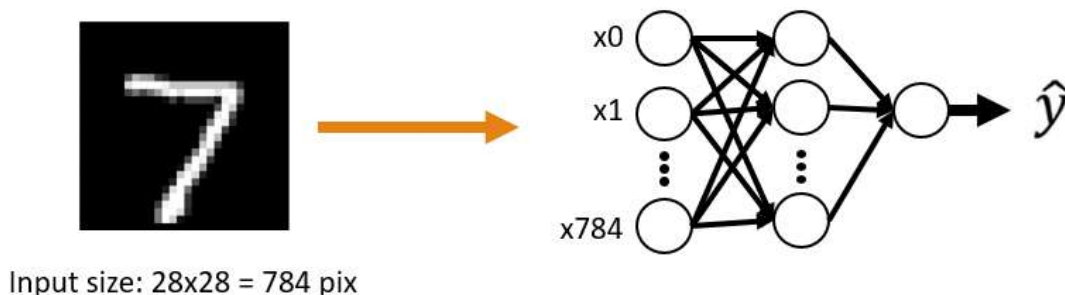
## 3. Classificação de Imagens e Redes Neurais Convolucionais

Esse capítulo tem como objetivo discutir o processo de classificação de imagens utilizando Redes Neurais Artificiais e comparar diferentes arquiteturas usando MLP (*Multilayer perceptron*) e CNN (*Convolutional Neural Network*).

### 3.1. Usando redes neurais artificiais para classificar imagens

O processo de classificação de uma imagem é categorizar todos os pixels da imagem em uma de várias classes. Nessa categorização é possível identificar quais objetos estão presentes na imagem através da extração de características. As áreas de aplicações de classificação de imagens são diversas e entre elas está o diagnóstico médico (LI, 2014).

Redes neurais artificiais têm sido campo de estudo para a resolução de problemas complexos de classificação de imagens (LECUN, 1989). A grande vantagem do uso de redes neurais artificiais é que o algoritmo pode ser generalizado para resolver diferentes problemas usando arquiteturas similares (LI, 2014).



**Figura 7.** Representação de uma arquitetura básica de uma rede neural convolucional para classificação de dígitos escritos.

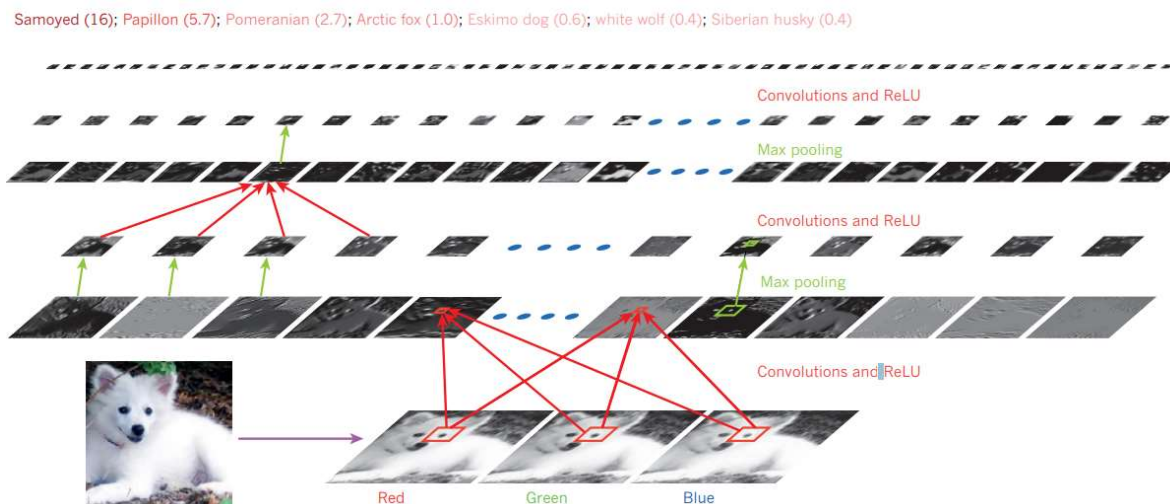
A Figura 7 representa a arquitetura MLP mais básica para reconhecimento de dígitos e que pode alcançar bons níveis de acurácia. Apesar do bom resultado, essa abordagem pode apresentar limitações e desvantagens quando aplicados para problemas mais complexos. Nessa rede, cada pixel da imagem servirá de entrada para um neurônio distinto na camada de entrada do modelo, nesse ponto, dois problemas principais dessa abordagem se tornam visíveis: A perda da informação espacial da imagem e a grande quantidade de parâmetros necessários que é proporcional ao tamanho da imagem.

Para contornar esses problemas uma diferente abordagem foi proposta por LeCun (1989) inspirada no funcionamento do córtex visual dos mamíferos. Ao aplicar filtros convolucionais para extrair as características da imagem, é possível manter a quantidade de parâmetros da rede independentemente do tamanho da entrada e manter a relação espacial dos filtros da imagem fazendo com que essa rede tenha performance superior em resolver muitos problemas de classificação complexos (LI, 2014).

## 3.2. Redes neurais convolucionais

As redes neurais convolucionais são arquiteturas de aprendizado profundo que subdividem os dados para tentar extrair características de cada conjunto (MIYAZAKI, 2017). Uma arquitetura típica de uma CNN (Figura 8) é estruturada como uma série de estágios. Os primeiros estágios são compostos de dois tipos de camadas: Camadas convolucionais e camadas *pooling* (LECUN, 2015). As camadas convolucionais são responsáveis pela extração das características enquanto as camadas de *pooling* são responsáveis por reduzir a dimensionalidade da informação, por fim, as camadas finais da rede são totalmente conectadas, semelhante a MLP.





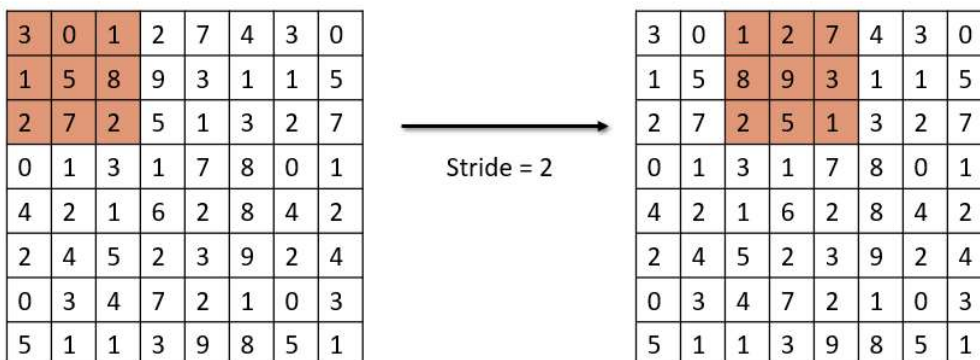
**Figura 8.** Representação da extração de características da convolução numa rede neural convolucional. (LECUN, 2015).

As camadas convolucionais são conjuntos de filtros não lineares que percorrem sequencialmente os dados de entrada (ou camada anterior) e então produzem matrizes chamadas mapas de características (*feature maps*) (MIYAZAKI, 2017). Durante o processo de treinamento, esses filtros são ajustados automaticamente para que sejam ativados na presença de características relevante, como orientação de bordas ou manchas de cores (KARPATHY, 2017). Em cada camada convolucional diversos filtros são usados, e os mapas de características produzidos são então empilhados, formando uma matriz 3D para imagens 2D.

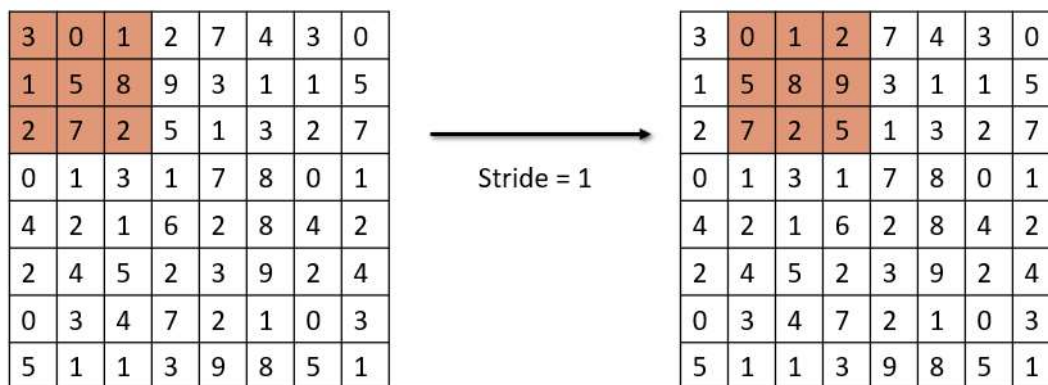
### 3.2.1. Passo de deslocamento das janelas convolucionais

Esse parâmetro, chamado por vezes de *stride*, determinam a forma como os filtros são aplicados na entrada. Conforme a Figura 9, esse parâmetro determina a distância do próximo passo de aplicação do filtro, quanto maior for, menor será o tamanho da saída, isso faz com que a camada convolucional além de extrair características da entrada também reduza sua dimensionalidade.

Stride



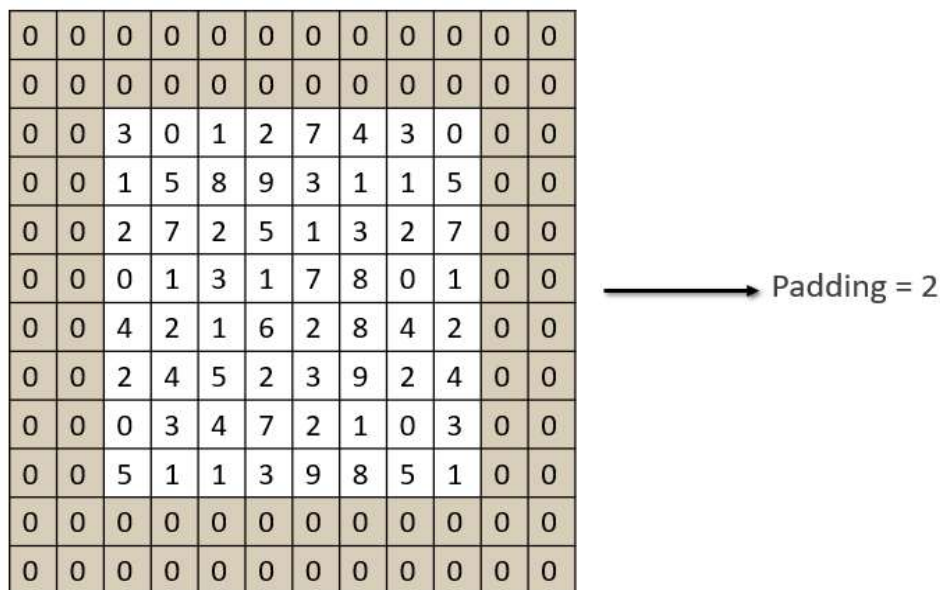
**Figura 9.** Representação da influência do hiperparâmetro *stride* na operação da convolução. Nessa imagem demonstra o deslocamento da janela convolucional para um valor de *stride* igual a dois.



**Figura 10.** Representação da influência do hiperparâmetro *stride* na operação da convolução. Nessa imagem demonstra o deslocamento da janela convolucional para um valor de *stride* igual a um.

### 3.2.2. Padding

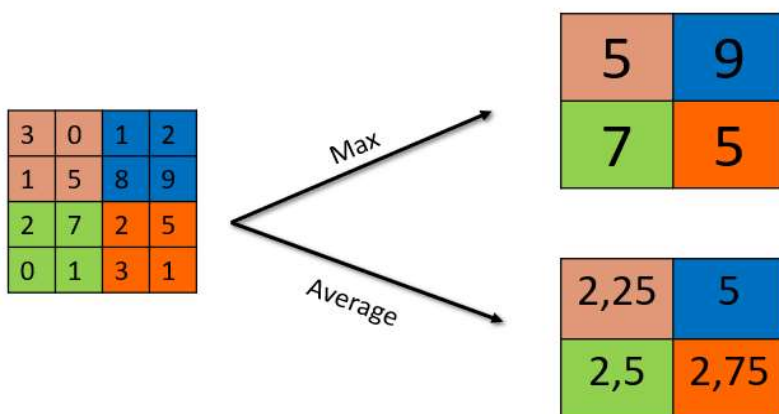
Observando as Figuras 9 e 10 é fácil observar que os filtros passarão pelos pixels da borda da imagem com menos frequência do que os pixels no meio da imagem. É possível controlar esse comportamento com o *padding* que basicamente é o preenchimento com zeros das bordas da imagem, com isso é possível controlar também o tamanho da saída da camada convolucional.



**Figura 11.** Imagem representada pela área branca preenchida com padding igual a 2 (área marrom).

### 3.2.3. Camada de *Pooling*

A camada de *pooling* que é usada tipicamente após uma camada convolucional, têm como objetivo reduzir a dimensão da camada de entrada, diminuindo o custo computacional e evitando o *overfitting* (MIYAZAKI, 2017). Para isso existem dois métodos de *pooling* que são o *Average* e o *Max Pooling*. O primeiro calcula a média dos valores da região de *pooling*, o segundo retorna o maior valor da região de *pooling*.



**Figura 12.** Representação das operações de Average e Max Pooling.

## 4. Experimentos

Este capítulo descreve os vários experimentos realizados para obter o modelo que apresenta o melhor desempenho na classificação de imagens obtidas no teste do desenho do pentágono realizado no MMSE.

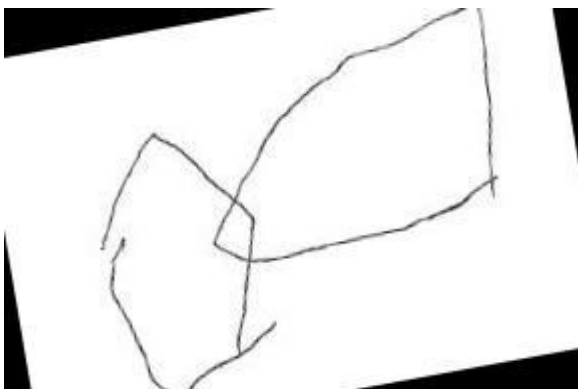
### 4.1. Base de Dados

A base de dados utilizada foi construída a partir de dados coletados de voluntários e expandida através do método de *data augmentation*. A tabela 1 demonstra a quantidade total de imagens disponíveis para os experimentos e a sua distribuição de acordo com a classe e divisão entre dados de treinamento e dados de teste.

Classe	Dados de treinamento	Dados de teste
Corretas	505	226
Incorretas	505	227
<b>Total</b>	<b>1010</b>	<b>253</b>

**Tabela 1.** Distribuição das bases de dados

A Figura 14 demonstra uma imagem retirada da base de dados obtida com o auxílio de voluntários que simularam o exame, a Figura 13 demonstra uma imagem obtida através de técnicas de *data augmentation*. Algumas imagens possuíam as bordas com manchas escuras, então foi criada uma segunda base de dados com as imagens pré-processada.



**Figura 13.** Imagem obtida com técnicas de *data augmentation*.



**Figura 14.** Imagem obtida com auxílio de voluntários.

Duas últimas bases de dados foram geradas a partir das duas primeiras com os dados normalizados, onde cada pixel que possui um valor no intervalo entre 0 e 255 na imagem original foi transformado em um pixel com o valor no intervalo em 0 e 1. Para realizarmos essa normalização basta dividirmos o valor de cada pixel ou valor na matriz que representa a imagem por 255.

Com isso a tabela 2 resume a organização da base de dados entre suas características, original, pré-processada ou normalizada.

Base de dados	Características	
1	Normalizada	Pré-processada
2	Não-normalizada	
3	Normalizada	Original
4	Não-normalizada	

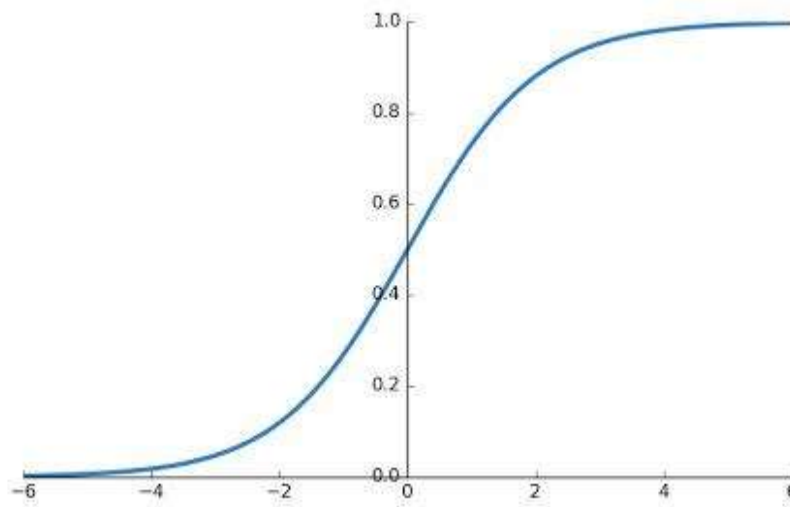
**Tabela 2.** Divisão das bases de dados

## 4.2. Treinamentos e testes

Para o treinamento de todos os modelos descritos nesta sessão, foi desenvolvido um script em *Python* e executado na plataforma *Google Colab*. Em todos os cenários o tamanho da imagem de entrada é 320x248.

### 4.2.1. Experimentos para validação da arquitetura da rede

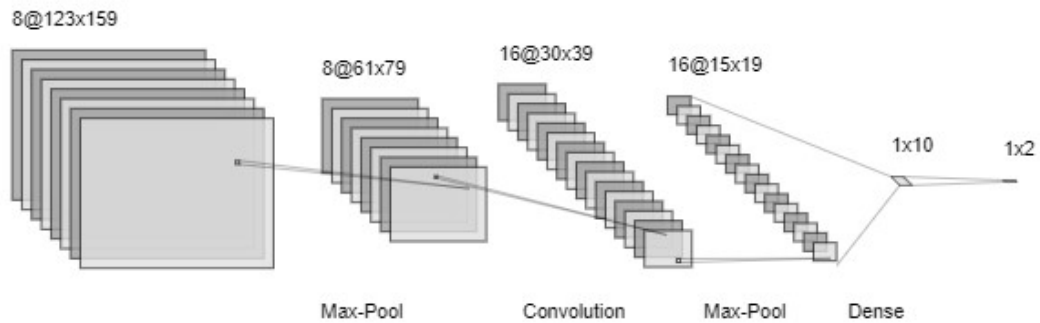
Os objetivos desses experimentos são encontrar qual arquitetura da rede apresenta melhor acurácia na classificação das imagens, entender também o impacto de possuir mais ou menos camadas convolucionais e mais ou menos camadas totalmente conectadas na rede. A base de dados utilizada foi a não pré-processada e não normalizada. Todos os experimentos deste item utilizaram a função de ativação *sigmoid* (Figura 15) em todas as camadas escondidas e a função *softmax*, definida por  $\delta(x_j) = \frac{e^{x_j}}{\sum_i e^{x_i}}$ , na camada de saída.



**Figura 15.** Função sigmoide

A escolha da função *sigmoid* é devido a sua não-linearidade que, através de experimentos, obteve resultados mais adequados para lidar com entradas com valores no intervalo entre 0 e 255. A função *softmax* é muito adequada quando utilizada como função de ativação da última camada em problemas de classificação pois, conforme sua definição, ela cria uma distribuição de probabilidade entre as saídas dos neurônios da camada de saída, logo, conforme a saída de um neurônio se aproxima de 1 as outras se aproximam de 0, com isso também é possível obter um maior aproveitamento do treinamento da rede.

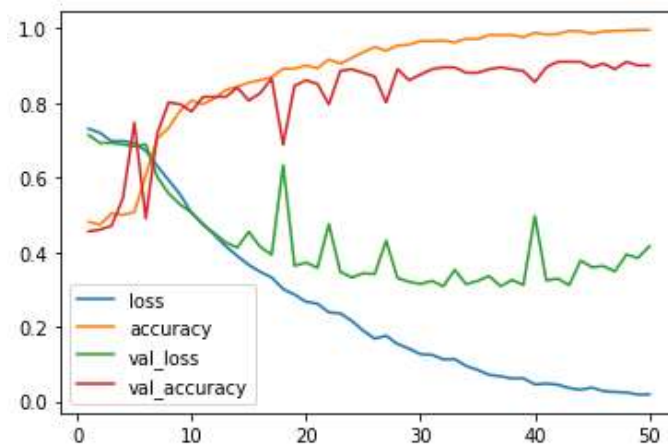
## Primeiro experimento



**Figura 16.** Diagrama da arquitetura aplicada no primeiro experimento do item 4.2.1.

Para esse experimento foi escolhida a função de ativação *sigmoid* em todas as camadas exceto a camada de saída que possui a função *softmax* como ativação. Logo, baseado na Figura 16 é possível descrever cada camada da rede desse experimento da seguinte forma:

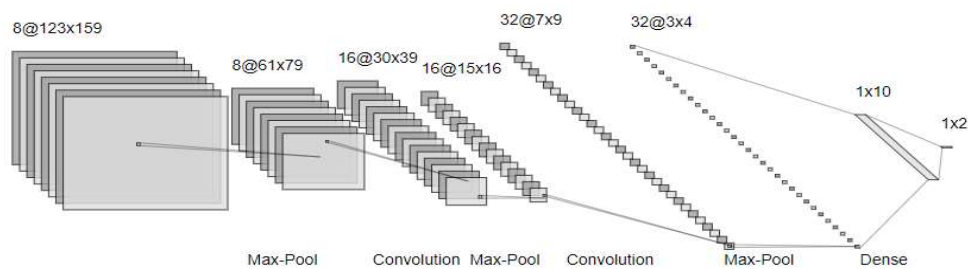
1. 8 Filtros Convolucionais tamanho 3x3; *Stride* 2
2. Camada de *pooling*; *Max Pooling* tamanho 2x2; *Stride* 2
3. 16 Filtros Convolucionais tamanho 3x3; *Stride* 2
4. Camada de *pooling*; *Max Pooling* tamanho 2x2; *Stride* 2
5. Camada totalmente conectada com 10 neurônios
6. Camada de saída totalmente conectada com 2 neurônios



**Figura 17.** Gráfico demonstrando desempenho da rede durante treinamento

A Figura 17 demonstra o gráfico com os valores de acurácia e erro desse experimento. Notar que as linhas de *loss* e *accuracy* representam respectivamente o erro e a acurácia para o conjunto de treinamento e as linhas *val\_loss* e *val\_accuracy* representam respectivamente o erro e a acurácia para o conjunto de validação do treinamento. Essa rede obteve valor do erro foi de 0.29 e o valor da acurácia para o conjunto de testes foi de 92,88%.

### Segundo experimento

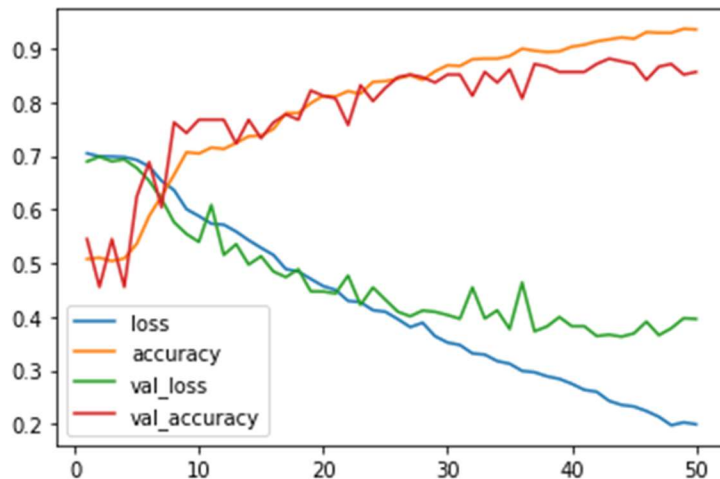


**Figura 18.** Diagrama da arquitetura aplicada no segundo experimento do item 4.2.1.

Nesse experimento foi utilizada uma arquitetura similar ao do experimento anterior, no entanto uma camada convolucional e uma camada de *pooling*, logo baseado na Figura 17 a rede possui as seguintes camadas:

1. 8 Filtros Convolucionais tamanho 3x3; *Stride* 2
2. Camada de *pooling*; *Max Pooling* tamanho 2x2; *Stride* 2
3. 16 Filtros Convolucionais tamanho 3x3; *Stride* 2
4. Camada de *pooling*; *Max Pooling* tamanho 2x2; *Stride* 2
5. 32 Filtros Convolucionais tamanho 3x3; *Stride* 2x2
6. Camada de *pooling*; *Max Pooling* tamanho 2x2; *Stride* 2
7. Camada totalmente conectada com 10 neurônios
8. Camada de saída totalmente conectada com 2 neurônios

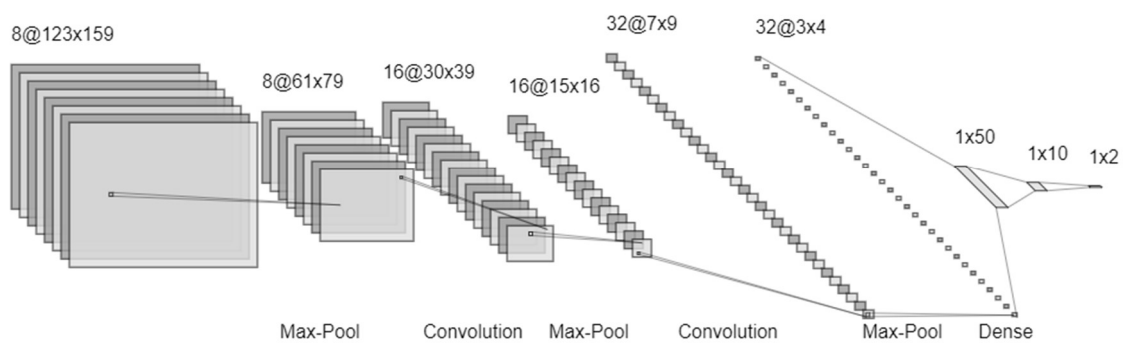




**Figura 19.** Gráfico demonstrando desempenho da rede do segundo experimento durante treinamento

A Figura 19 demonstra o gráfico com os valores de acurácia e erro desse experimento. Notar que as linhas de *loss* e *accuracy* representam respectivamente o erro e a acurácia para o conjunto de treinamento e as linhas *val\_loss* e *val\_accuracy* representam respectivamente o erro e a acurácia para o conjunto de validação do treinamento. Essa rede obteve valor do erro foi de 0.26 e o valor da acurácia para o conjunto de testes foi de 91,30%.

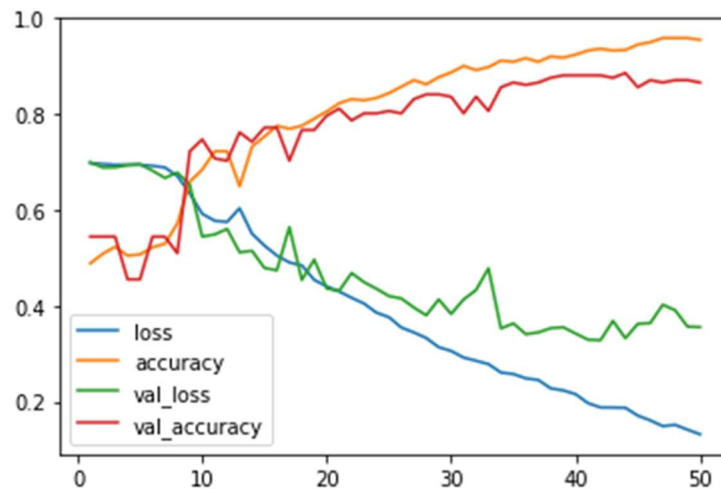
### Terceiro experimento



**Figura 20.** Diagrama da arquitetura aplicada no terceiro experimento do item 4.2.1.

Nesse experimento foi utilizada uma arquitetura similar ao do experimento anterior, no entanto foi adicionada uma camada convolucional e uma camada de pooling, logo baseado na Figura 20 a rede possui as seguintes camadas:

1. 8 Filtros Convolucionais tamanho 3x3; *Stride* 2
2. Camada de *pooling*; *Max Pooling* tamanho 2x2; *Stride* 2
3. 16 Filtros Convolucionais tamanho 3x3; *Stride* 2
4. Camada de *pooling*; *Max Pooling* tamanho 2x2; *Stride* 2
5. 32 Filtros Convolucionais tamanho 3x3; *Stride* 2
6. Camada de *pooling*; *Max Pooling* tamanho 2x2; *Stride* 2
7. Camada totalmente conectada com 50 neurônios
8. Camada totalmente conectada com 10 neurônios
9. Camada de saída totalmente conectada com 2 neurônios



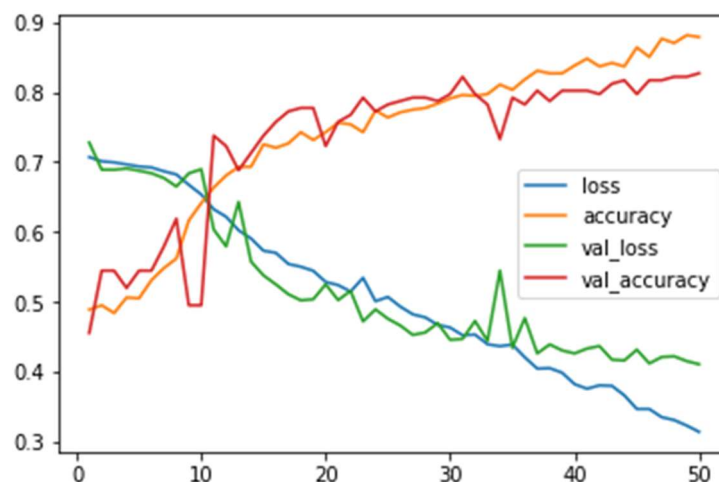
**Figura 21.** Gráfico que demonstra desempenho da rede durante o treinamento do terceiro experimento

A Figura 21 demonstra o gráfico com os valores de acurácia e erro desse experimento. Notar que as linhas de *loss* e *accuracy* representam respectivamente o erro e a acurácia para o conjunto de treinamento e as linhas *val\_loss* e *val\_accuracy* representam respectivamente o erro e a acurácia para o conjunto de validação do treinamento. Essa rede obteve valor do erro foi de 0.30 e o valor da acurácia para o conjunto de testes foi de 87,35%.

### Quarto experimento

Nesse experimento objetivo é compreender o impacto da alteração do valor do *stride* nas camadas de pooling. Utilizando a arquitetura descrita no primeiro experimento, verificar Figura 16, foi aumentado o valor do *stride* das camadas de *pooling* para 4, dessa forma a rede possui as seguintes camadas:

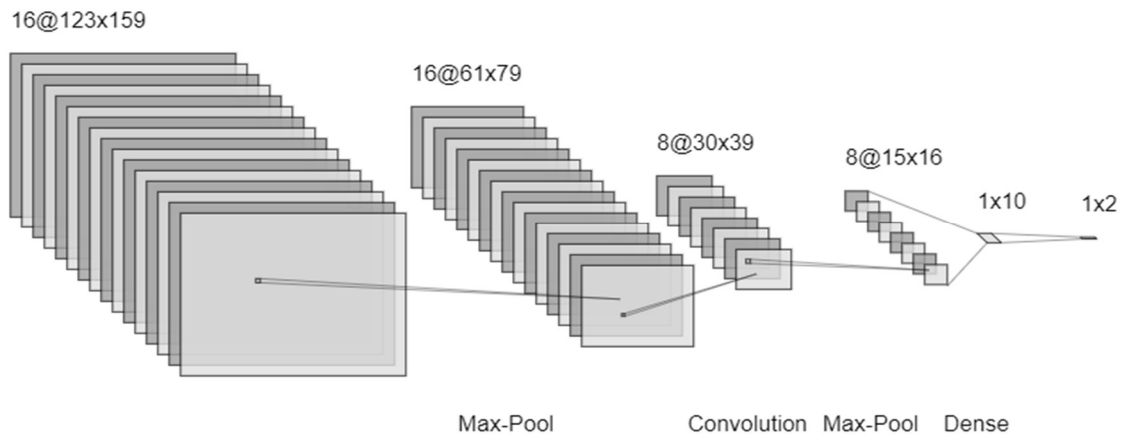
1. 8 Filtros Convolucionais tamanho 3x3; *Stride* 2
2. Camada de *pooling*; *Max Pooling* tamanho 2x2; *Stride* 4
3. 16 Filtros Convolucionais tamanho 3x3; *Stride* 2
4. Camada de *pooling*; *Max Pooling* tamanho 2x2; *Stride* 4
5. Camada totalmente conectada com 10 neurônios
6. Camada de saída totalmente conectada com 2 neurônios



**Figura 22.** Gráfico que demonstra desempenho da rede no quarto experimento do item 4.2.1

A Figura 22 demonstra o gráfico com os valores de acurácia e erro desse experimento. Notar que as linhas de *loss* e *accuracy* representam respectivamente o erro e a acurácia para o conjunto de treinamento e as linhas *val\_loss* e *val\_accuracy* representam respectivamente o erro e a acurácia para o conjunto de validação do treinamento. Essa rede obteve valor do erro foi de 0.37 e o valor da acurácia para o conjunto de testes foi de 84,18%.

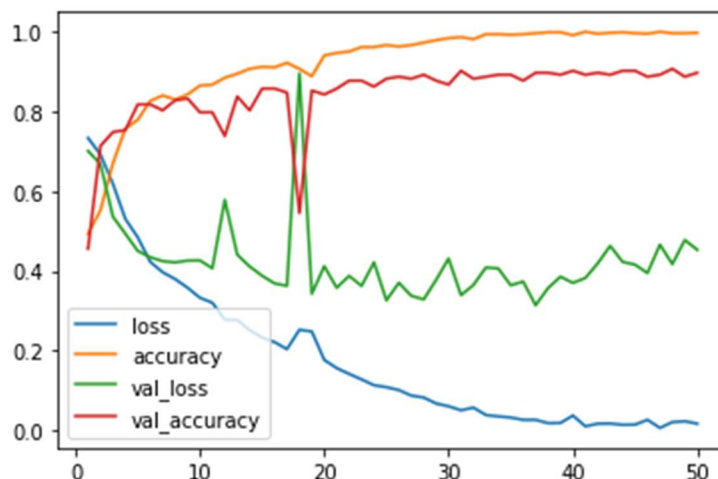
### Quinto experimento



**Figura 23.** Diagrama da arquitetura aplicada no quinto experimento do item 4.2.1.

Nesse experimento o objetivo é compreender o efeito de ter mais filtros nas primeiras camadas convolucionais, então foi invertido o esquema da primeira arquitetura. Logo a rede possui as seguintes camadas:

1. 16 Filtros Convolucionais tamanho 3x3; *Stride* 2
2. Camada de *pooling*; *Max Pooling* tamanho 2x2; *Stride* 2
3. 8 Filtros Convolucionais tamanho 3x3; *Stride* 2
4. Camada de *pooling*; *Max Pooling* tamanho 2x2; *Stride* 2
5. Camada totalmente conectada com 10 neurônios
6. Camada de saída totalmente conectada com 2 neurônios

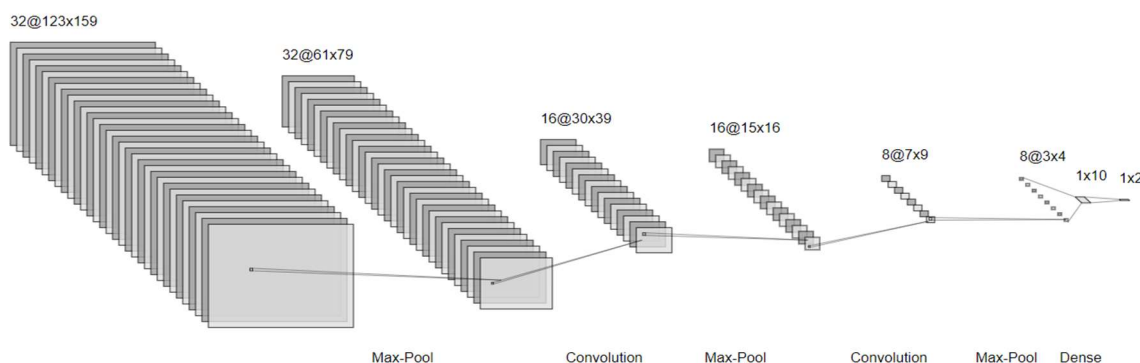


**Figura 24.** Gráfico exibindo desempenho da rede durante o treinamento do quinto experimento.

A Figura 24 demonstra o gráfico com os valores de acurácia e erro desse experimento. Notar que as linhas de *loss* e *accuracy* representam respectivamente o erro e a acurácia para o conjunto de treinamento e as linhas *val\_loss* e *val\_accuracy* representam respectivamente o erro e a acurácia para o conjunto de validação do treinamento. Essa rede obteve valor do erro foi de 0.34 e o valor da acurácia para o conjunto de testes foi de 92,09%.

### Sexto experimento

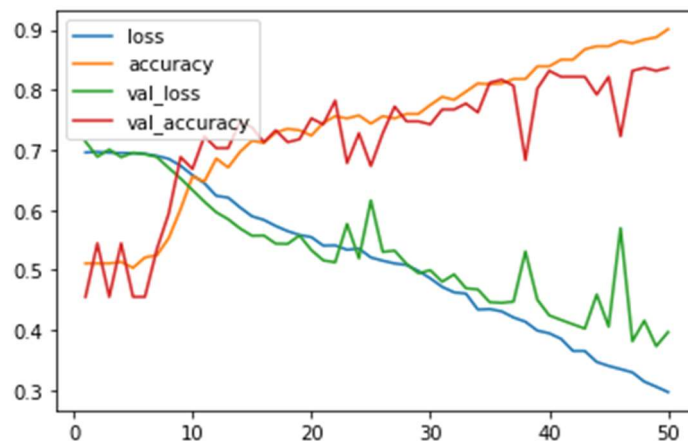
A ideia desse experimento é realizar a mesma validação do experimento anterior, porém utilizar uma arquitetura com mais camadas convolucionais.



**Figura 25.** Diagrama da arquitetura aplicada no sexto experimento do item 4.2.1

Com isso o modelo possui seguintes camadas:

1. 32 Filtros Convolucionais tamanho 3x3; *Stride* 2
2. Camada de pooling; Max Pooling tamanho 2x2; *Stride* 2
3. 16 Filtros Convolucionais tamanho 3x3; *Stride* 2
4. Camada de pooling; Max Pooling tamanho 2x2; *Stride* 2
5. 8 Filtros Convolucionais tamanho 3x3; *Stride* 2
6. Camada de pooling; Max Pooling tamanho 2x2; *Stride* 2
7. Camada totalmente conectada com 50 neurônios
8. Camada totalmente conectada com 10 neurônios
9. Camada de saída totalmente conectada com 2 neurônios



**Figura 26.** Gráfico demonstrando o desempenho da rede do sexto experimento do item 4.2.1.

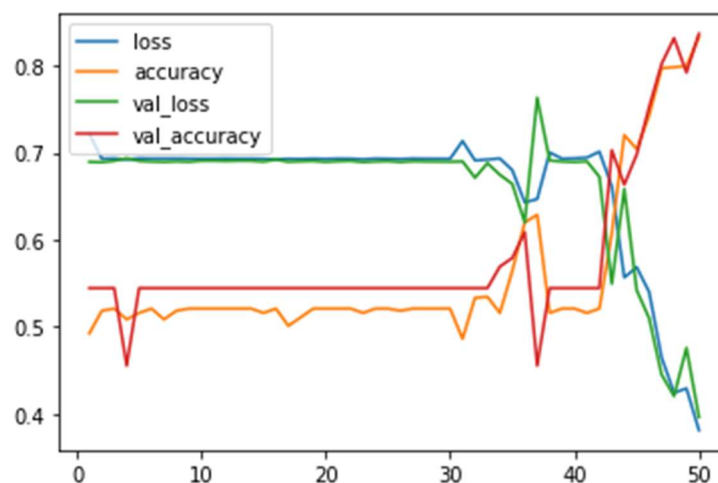
A Figura 26 demonstra o gráfico com os valores de acurácia e erro desse experimento. Notar que as linhas de *loss* e *accuracy* representam respectivamente o erro e a acurácia para o conjunto de treinamento e as linhas *val\_loss* e *val\_accuracy* representam respectivamente o erro e a acurácia para o conjunto de validação do treinamento. Essa rede obteve valor do erro foi de 0.33 e o valor da acurácia para o conjunto de testes foi de 86,16%.

### Sétimo experimento

Por último o objetivo é compreender o efeito do aumento do tamanho do filtro convolucional na primeira camada, para isso foi utilizada a primeira arquitetura (ver

Figura 16) e alterado o tamanho do filtro na primeira camada convolucional. Com isso a rede possui as seguintes camadas:

1. 8 Filtros Convolucionais tamanho 5x5; *Stride* 2
2. Camada de *pooling*; *Max Pooling* tamanho 2x2; *Stride* 2
3. 16 Filtros Convolucionais tamanho 3x3; *Stride* 2
4. Camada de *pooling*; *Max Pooling* tamanho 2x2; *Stride* 2
5. Camada totalmente conectada com 10 neurônios
6. Camada de saída totalmente conectada com 2 neurônios



**Figura 27.** Gráfico que demonstra desempenho do treinamento da rede no sétimo experimento do item 4.2.1.

A Figura 27 demonstra o gráfico com os valores de acurácia e erro desse experimento. Notar que as linhas de *loss* e *accuracy* representam respectivamente o erro e a acurácia para o conjunto de treinamento e as linhas *val\_loss* e *val\_accuracy* representam respectivamente o erro e a acurácia para o conjunto de validação do treinamento. É importante observar que durante mais da metade do treinamento os valores do erro e acurácia se mantiveram próximos de 0,40 e 0,70 respectivamente, o que demonstra que a rede durante essas épocas de treinamento estava presa em um mínimo local e não conseguiram aprender com o conjunto de treinamento. Após a época de número 40 é possível observar uma convergência dos pesos da rede. Essa rede obteve valor do erro foi de 0.32 e o valor da acurácia para o conjunto de testes foi de 86,95%.

## Resumo

Como resultado é possível concluir que as melhores arquiteturas para classificação da base de dados não pré-processada e não normalizada são as utilizadas nos primeiro e segundo experimentos. No geral a tabela 3 resume os resultados obtidos em todos os experimentos deste item.

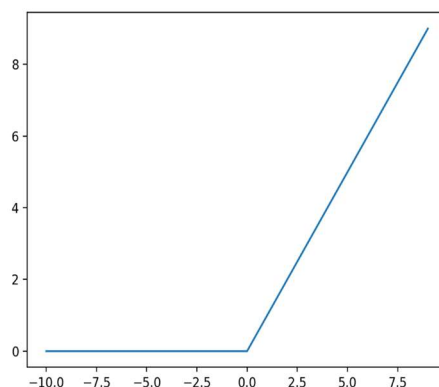
Número experimento	Quantidade camadas convolucionais	Quantidade camadas totalmente conectadas	Erro apresentado no conjunto de teste	Acurácia apresentada no conjunto de teste
1	2	2	0,29	92,88%
2	3	2	0,26	91,30%
3	3	3	0,30	87,35%
4	2	2	0,37	84,18%
5	2	2	0,34	92,09%
6	3	2	0,33	86,16%
7	2	2	0,32	86,95%

**Tabela 3.** Resumo dos resultados obtidos nos experimentos do item 4.2.1.

### 4.2.2. Experimentos com bases normalizadas

Nesses experimentos foram utilizadas as duas arquiteturas com melhor desempenho de acordo com a tabela 3 e treinada com a base de dados normalizada não pré-processada. Também foi analisado o impacto da modificação de alguns hiperparâmetros da rede e da utilização de diferentes otimizadores no resultado do treinamento. Ao normalizar as imagens para que os pixels estejam no intervalo entre 0 e 1, a função *sigmoid* se mostrou inadequada para utilização nas camadas escondidas, por isso é a função ReLU (*Rectified Linear Unit*) (Figura 28) mostrou o melhor desempenho durante os experimentos e foi utilizada.

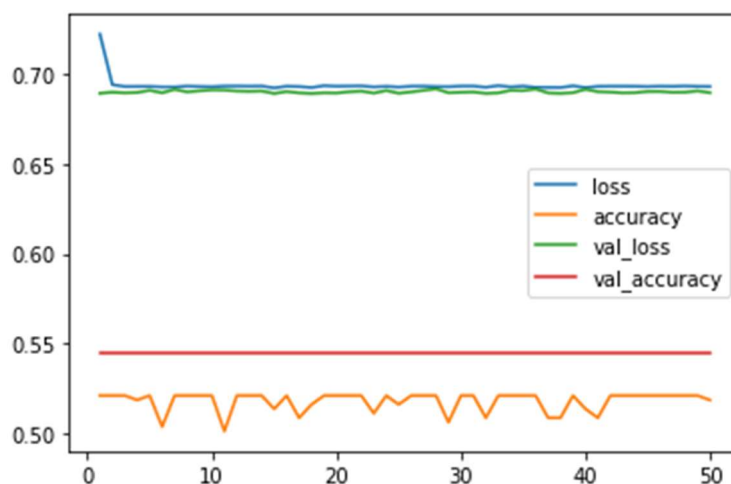




**Figura 28.** Função ReLU.

### Primeiro Experimento

O objetivo desse experimento é repetir o primeiro experimento do item 4.2.1. porém utilizando a base de dados normalizada – os valores dos pixels das imagens se encontram no intervalo entre 0 e 1.



**Figura 29.** Gráfico que demonstra desempenho da rede durante treinamento no primeiro experimento do item 4.2.2.

A Figura 29 demonstra o gráfico com os valores de acurácia e erro para experimento. Notar que as linhas de *loss* e *accuracy* representam respectivamente o erro e a acurácia para o conjunto de treinamento e as linhas *val\_loss* e *val\_accuracy* representam respectivamente o erro e a acurácia para o conjunto de validação do treinamento.

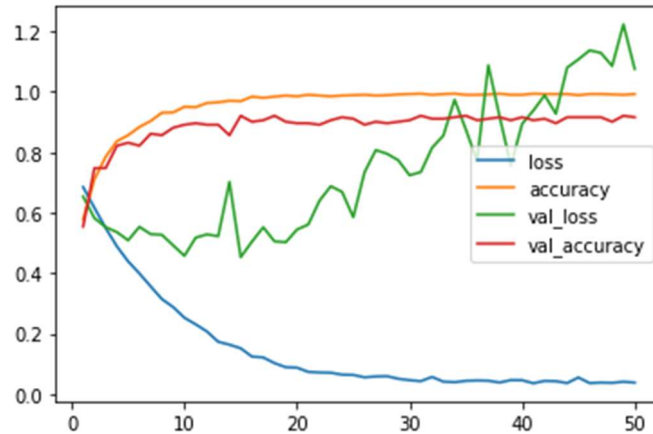
Analisando o gráfico de convergência da Figura 29 é possível analisar que a rede não conseguiu convergir para algum resultado. Essa rede obteve valor do erro foi de 0,68 e o valor da acurácia para o conjunto de testes foi de 56,52%.

### **Segundo experimento**

Nesse primeiro experimento foi utilizada a arquitetura definida no primeiro experimento do item 4.2.1 (ver Figura 16) com a base de dados normalizada e sem pré-processamento. A Figura 30 demonstra o gráfico com os valores de acurácia e erro para experimento. Notar que as linhas de *loss* e *accuracy* representam respectivamente o erro e a acurácia para o conjunto de treinamento e as linhas *val\_loss* e *val\_accuracy* representam respectivamente o erro e a acurácia para o conjunto de validação do treinamento.

Analisando o gráfico de convergência da figura 30 é possível notar que a rede consegue convergir para algum resultado porém apenas até aproximadamente a rodada de treinamento de número 20 é possível notar que a rede está aprendendo – notar que o valor do erro para os dados de validação até a época de número 20 está decrescendo. A partir daí a rede não consegue mais aprender com o conjunto de treinamento – notar o valor de erro ascendendo a partir da época de número 20.

Essa rede obteve valor do erro foi de 0,42 e o valor da acurácia para o conjunto de testes foi de 93,28%. Após executar o mesmo experimento para as bases de dados normalizadas e não normalizadas, concluímos que para os valores não normalizados a função de ativação mais adequada para as camadas escondidas é a *sigmoid* já para os valores normalizados a função de ativação mais adequada é a *ReLU*.

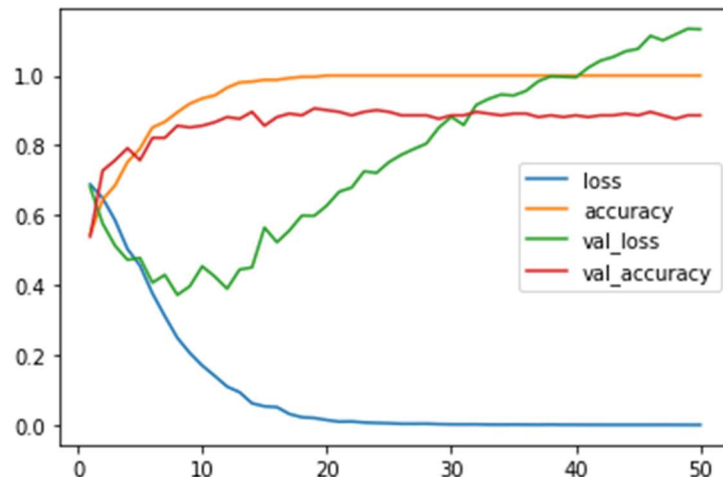


**Figura 30.** Gráfico que demonstra desempenho da rede no segundo experimento do item 4.2.2.

### Terceiro experimento

Nesse experimento foi utilizada a arquitetura definida no segundo experimento do item 4.2.1 (ver Figura 18) com a base de dados normalizada e sem pré-processamento. A Figura 31 demonstra o gráfico com os valores de acurácia e erro para experimento. Notar que as linhas de *loss* e *accuracy* representam respectivamente o erro e a acurácia para o conjunto de treinamento e as linhas *val\_loss* e *val\_accuracy* representam respectivamente o erro e a acurácia para o conjunto de validação do treinamento.

Observar que, semelhante ao experimento anterior, essa rede só conseguiu aprender com o conjunto de treinamento até a rodada de testes de número 20, como o valor do erro para o conjunto de validação está crescendo isso significa que a rede não estava conseguindo aprender. Essa rede obteve valor do erro foi de 0,42 e o valor da acurácia para o conjunto de testes foi de 93,28%.



**Figura 31.** Gráfico que demonstra desempenho da rede no terceiro experimento do item 4.2.2.

### 4.2.3. Experimentos com a base normalizada e pré-processada

Nos próximos experimentos foram definidas novas arquiteturas, porém semelhantes às que tiveram resultados melhores no item 4.2.1 e aplicada no base de dados normalizada e pré-processada. Foram utilizadas as funções ReLU, para as camadas escondidas, uma vez que estamos utilizando valores normalizados – no intervalo entre 0 e 1. A função *softmax* foi escolhida como ativação para a camada de saída.

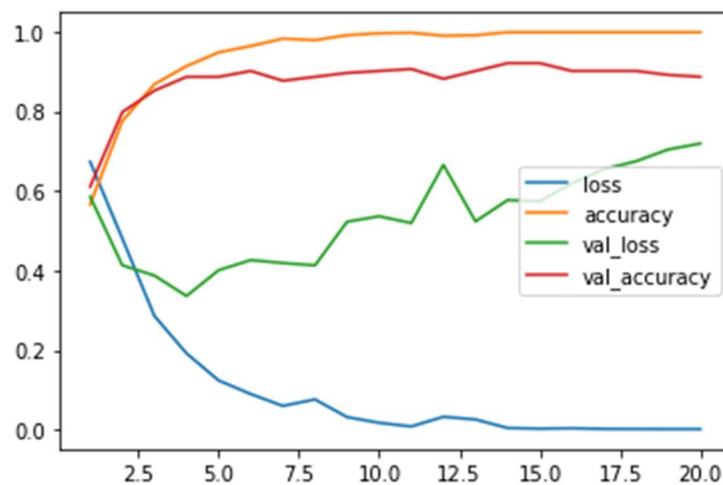
Como aprendizado dos experimentos do item 4.2.2. como foram utilizadas a base normalizada e a função ReLU nas camadas escondidas, foi alterado o limite de rodadas de treinamento de 20 para 50, dessa forma a expectativa é executar o treinamento apenas até onde a rede consegue convergir para algum resultado.

#### Primeiro experimento

Foi definida uma arquitetura conforme Figura 18, porém foi alterado o valor do *stride* das camadas convolucionais para 1, logo a rede possui as seguintes camadas:

1. 8 Filtros Convolucionais tamanho 3x3; *Stride* 1
2. Camada de *pooling*; *Max Pooling* tamanho 2x2; *Stride* 2

3. 16 Filtros Convolucionais tamanho 3x3; *Stride* 1
4. Camada de *pooling*; *Max Pooling* tamanho 2x2; *Stride* 2
5. 32 Filtros Convolucionais tamanho 3x3; *Stride* 1
6. Camada de *pooling*; *Max Pooling* tamanho 2x2; *Stride* 2
7. Camada totalmente conectada com 10 neurônios
8. Camada de saída totalmente conectada com 2 neurônios



**Figura 32.** Gráfico que demonstra desempenho do treinamento da rede no primeiro experimento do item 4.2.3.

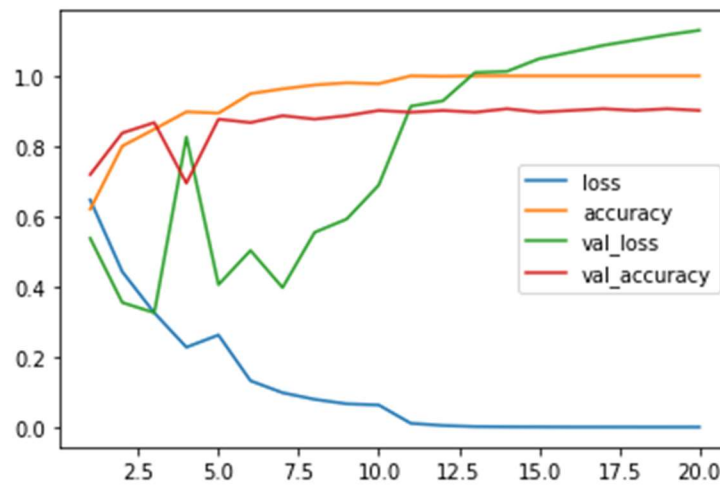
A Figura 32 demonstra o gráfico com os valores de acurácia e erro para experimento. Notar que as linhas de *loss* e *accuracy* representam respectivamente o erro e a acurácia para o conjunto de treinamento e as linhas *val\_loss* e *val\_accuracy* representam respectivamente o erro e a acurácia para o conjunto de validação do treinamento. Essa rede obteve valor do erro foi de 0,36 e o valor da acurácia para o conjunto de testes foi de 93,30%.

### Segundo experimento

Foi definida uma arquitetura conforme Figura 20, porém foi alterado o valor do *stride* das camadas convolucionais para 1, logo a rede possui as seguintes camadas:

1. 8 Filtros Convolucionais tamanho 3x3; *Stride* 1
2. Camada de *pooling*; *Max Pooling* tamanho 2x2; *Stride* 2

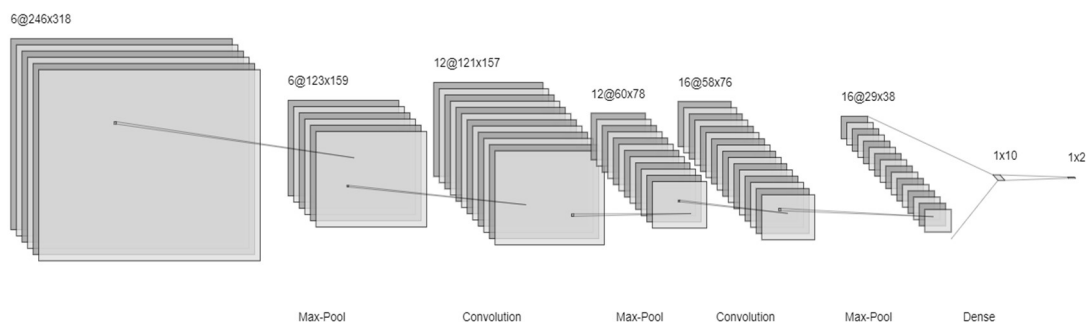
3. 16 Filtros Convolucionais tamanho 3x3; *Stride* 1
4. Camada de *pooling*; *Max Pooling* tamanho 2x2; *Stride* 2
5. 32 Filtros Convolucionais tamanho 3x3; *Stride* 1
6. Camada de *pooling*; *Max Pooling* tamanho 2x2; *Stride* 2
7. Camada totalmente conectada com 50 neurônios
8. Camada totalmente conectada com 10 neurônios
9. Camada de saída totalmente conectada com 2 neurônios



**Figura 33.** Gráfico que demonstra desempenho da rede durante o treinamento do segundo experimento do item 4.2.3.

A Figura 33 demonstra o gráfico com os valores de acurácia e erro para experimento. Notar que as linhas de *loss* e *accuracy* representam respectivamente o erro e a acurácia para o conjunto de treinamento e as linhas *val\_loss* e *val\_accuracy* representam respectivamente o erro e a acurácia para o conjunto de validação do treinamento. O valor do erro para o conjunto de teste foi de 0,29 e a acurácia nesse mesmo conjunto foi de 96,06%.

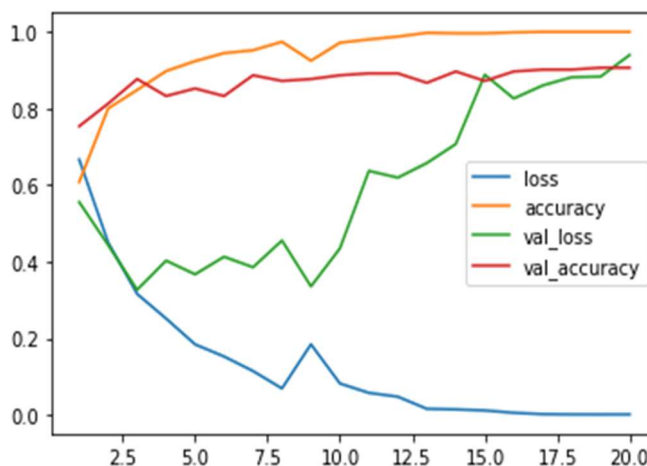
### Terceiro experimento



**Figura 34.** Diagrama da arquitetura aplicada no terceiro experimento do item 4.2.3.

Foi definida nova arquitetura baseada na Figura 34, porém o objetivo é analisar o impacto da diminuição da quantidade de filtros em todas as camadas convolucionais, com isso a rede possui as seguintes camadas:

1. 6 Filtros Convolucionais tamanho 3x3; *Stride* 1
2. Camada de *pooling*; *Max Pooling* tamanho 2x2; *Stride* 2
3. 12 Filtros Convolucionais tamanho 3x3; *Stride* 1
4. Camada de *pooling*; *Max Pooling* tamanho 2x2; *Stride* 2
5. 16 Filtros Convolucionais tamanho 3x3; *Stride* 1
6. Camada de *pooling*; *Max Pooling* tamanho 2x2; *Stride* 2
7. Camada totalmente conectada com 10 neurônios
8. Camada de saída totalmente conectada com 2 neurônios



**Figura 35.** Gráfico que demonstra desempenho da rede durante treinamento do terceiro experimento do item 4.2.3.

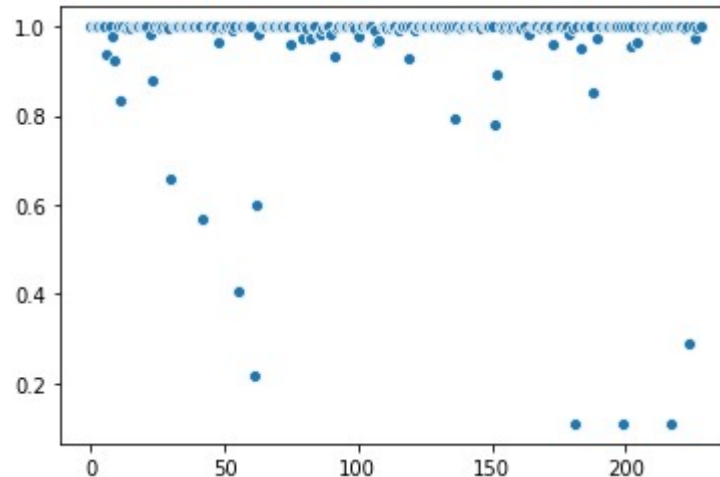
A Figura 35 demonstra o gráfico com os valores de acurácia e erro para experimento. Notar que as linhas de *loss* e *accuracy* representam respectivamente o erro e a acurácia para o conjunto de treinamento e as linhas *val\_loss* e *val\_accuracy* representam respectivamente o erro e a acurácia para o conjunto de validação do treinamento. O valor do erro para o conjunto de teste foi de 0,31 e a acurácia nesse mesmo conjunto foi de 94,48%.

### 4.3. Análise da certeza da rede

Para melhor entendimento do resultado da rede é necessário analisar a distribuição da resposta da rede, em todos os experimentos foi utilizado 2 saídas cada uma representando uma classe, caso a classificação para a imagem seja correta, a saída do modelo deverá ser [0;1] e [1;0] caso contrário. Como a rede dificilmente trará o resultado exato, é buscado sempre o índice do maior valor, logo, para a imagem correta a resposta deve ser 1 e para a imagem incorreta a imagem deve ser 0.

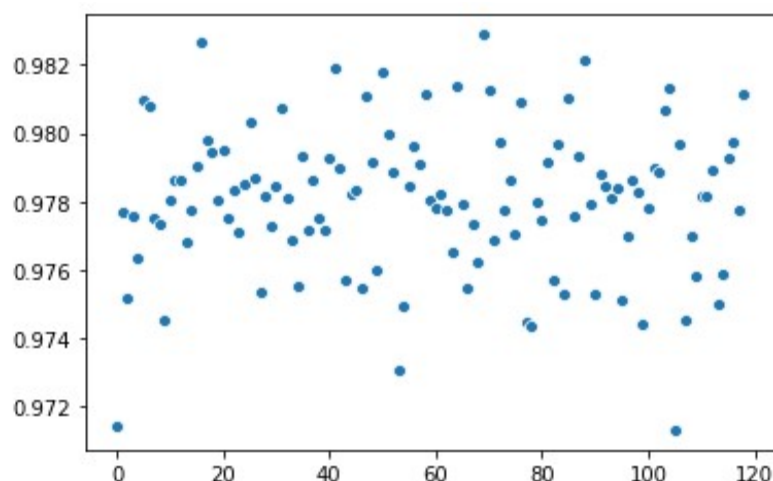
Com isso é possível plotar um gráfico contendo a destruição da distância das respostas da rede para o conjunto de teste e assim analisar o grau de certeza que o modelo está classificando as imagens. Logo, para as imagens não processadas e considerando a arquitetura da Figura 20, logo:





**Figura 36.** Distribuição da certeza para imagens classificadas corretamente, quanto mais próximo de 1, melhor. Base de dados não pré-processadas.

A Figura 36 demonstra que a maior parte das imagens, na base de dados não pré-processada, classificadas corretamente pelo modelo são bem próximas de  $[0;1]$  e  $[1;0]$ , no entanto, ainda existem algumas imagens que estão sendo classificadas com pouca certeza, por exemplo  $[0,5;0,4]$  para uma imagem incorreta. Porém ao analisarmos a mesma distribuição (Figura 37) para a base de dados pré-processada é verificado que todos os resultados corretos estão próximos de 1, com isso é possível concluir que as manchas pretas na imagem original acabam ocasionando ruídos e prejudicando a classificação com maior grau de certeza.



**Figura 37.** Distribuição da certeza para imagens classificadas corretamente, quanto mais próximo de 1, melhor. Base de dados pré-processadas.

## 4.4. Resumo dos resultados dos experimentos

Essa subseção apresenta de forma resumida os resultados dos experimentos realizados no item 4.2. A Tabela 4 apresenta o resumo de todos os experimentos realizados exibindo o valor do erro e da acurácia, para o conjunto de teste, de acordo com a base de dados utilizada e a configuração do modelo.

Base de dados	Configuração do modelo	Função ativação	Valor erro	Valor acurácia
Não normalizada e não processada	8 Conv.; 3x3; <i>Stride</i> 2 <i>Max-Pooling</i> 2x2 16 Conv.; 3x3; <i>Stride</i> 2 <i>Max-Pooling</i> 2x2 Densa; 10 neurônios Densa; 2 neurônios	Sigmoid	0,29	92,88%
Não normalizada e não processada	8 Conv.; 3x3; <i>Stride</i> 2 <i>Max-Pooling</i> 2x2 16 Conv.; 3x3; <i>Stride</i> 2 <i>Max-Pooling</i> 2x2 32 Conv.; 3x3; <i>Stride</i> 2 <i>Max-Pooling</i> 2x2 Densa; 10 neurônios Densa; 2 neurônios	Sigmoid	0,29	91,30%
Não normalizada e não processada	8 Conv.; 3x3; <i>Stride</i> 2 <i>Max-Pooling</i> 2x2 16 Conv.; 3x3; <i>Stride</i> 2 <i>Max-Pooling</i> 2x2 32 Conv.; 3x3; <i>Stride</i> 2 <i>Max-Pooling</i> 2x2 Densa; 50 Densa; 10 Densa; 2	Sigmoid	0,30	87,35%
Não normalizada e não processada	8 Conv.; 3x3; <i>Stride</i> 4 <i>Max-Pooling</i> 2x2 16 Conv.; 3x3; <i>Stride</i> 4 <i>Max-Pooling</i> 2x2 Densa; 10 neurônios Densa; 2 neurônios	Sigmoid	0,37	84,18%

Não normalizada e não processada	16 Conv.; 3x3; <i>Stride</i> 2 <i>Max-Pooling</i> 2x2 8 Conv.; 3x3; <i>Stride</i> 2 <i>Max-Pooling</i> 2x2 Densa; 10 neurônios Densa; 2 neurônios	Sigmoid	0,34	92,09%
Não normalizada e não processada	32 Conv.; 3x3; <i>Stride</i> 2 <i>Max-Pooling</i> 2x2 16 Conv.; 3x3; <i>Stride</i> 2 <i>Max-Pooling</i> 2x2 8 Conv.; 3x3; <i>Stride</i> 2 <i>Max-Pooling</i> 2x2 Densa; 50 Densa; 10 Densa; 2	Sigmoid	0,33	86,16%
Não normalizada e não processada	8 Conv.; 5x5; <i>Stride</i> 2 <i>Max-Pooling</i> 2x2 16 Conv.; 3x3; <i>Stride</i> 2 <i>Max-Pooling</i> 2x2 Densa; 10 neurônios Densa; 2 neurônios	Sigmoid	0,32	86,95%
Normalizada e não processada	8 Conv.; 3x3; <i>Stride</i> 2 <i>Max-Pooling</i> 2x2 16 Conv.; 3x3; <i>Stride</i> 2 <i>Max-Pooling</i> 2x2 Densa; 10 neurônios Densa; 2 neurônios	Sigmoid	0,68	56,52%
Normalizada e não processada	8 Conv.; 3x3; <i>Stride</i> 2 <i>Max-Pooling</i> 2x2 16 Conv.; 3x3; <i>Stride</i> 2 <i>Max-Pooling</i> 2x2 Densa; 10 neurônios Densa; 2 neurônios	ReLU	0,42	93,28%
Normalizada e não processada	8 Conv.; 3x3; <i>Stride</i> 2 <i>Max-Pooling</i> 2x2 16 Conv.; 3x3; <i>Stride</i> 2 <i>Max-Pooling</i> 2x2 32 Conv.; 3x3; <i>Stride</i> 2 <i>Max-Pooling</i> 2x2 Densa; 10 neurônios Densa; 2 neurônios	ReLU	0,64	90,09%

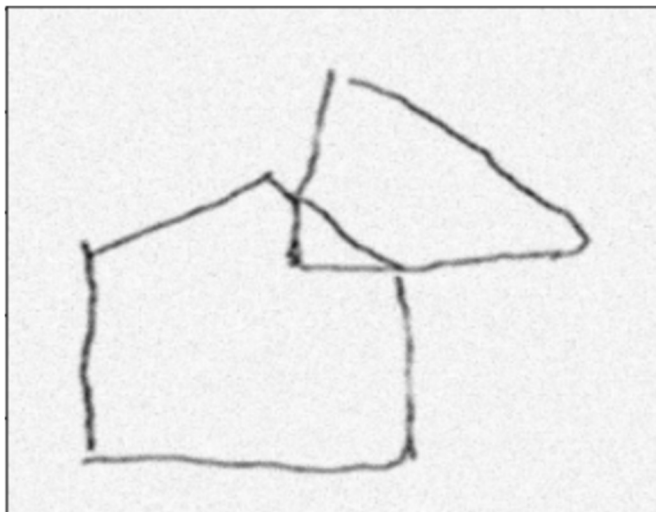
Normalizada e Processada	8 Conv.; 3x3; <i>Stride</i> 1 <i>Max-Pooling</i> 2x2 16 Conv.; 3x3; <i>Stride</i> 1 <i>Max-Pooling</i> 2x2 32 Conv.; 3x3; <i>Stride</i> 1 <i>Max-Pooling</i> 2x2 Densa; 10 neurônios Densa; 2 neurônios	ReLU	0,36	93,30
Normalizada e Processada	8 Conv.; 3x3; <i>Stride</i> 1 <i>Max-Pooling</i> 2x2 16 Conv.; 3x3; <i>Stride</i> 1 <i>Max-Pooling</i> 2x2 32 Conv.; 3x3; <i>Stride</i> 1 <i>Max-Pooling</i> 2x2 Densa; 50 Densa; 10 Densa; 2	ReLU	0,29	96,06%
Normalizada e Processada	6 Conv.; 3x3; <i>Stride</i> 1 <i>Max-Pooling</i> 2x2 12 Conv.; 3x3; <i>Stride</i> 1 <i>Max-Pooling</i> 2x2 16 Conv.; 3x3; <i>Stride</i> 1 <i>Max-Pooling</i> 2x2 Densa; 50 Densa; 10 Densa; 2	ReLU	0,31	94,48%

**Tabela 4.** Resumo dos resultados dos experimentos realizados no item 4.2

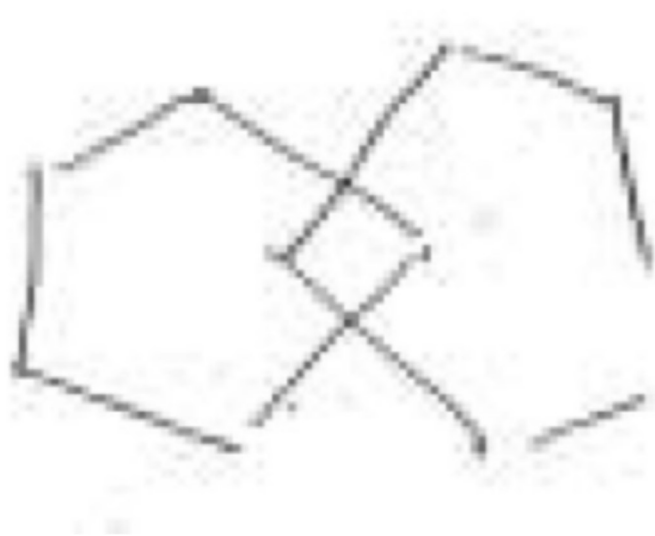
## 5. Resultados e conclusões

A automatização de testes clínicos para diagnóstico de demência oferecerá grande benefício a sociedade uma vez que facilitará o acesso dos pacientes a procedimentos considerados padrão-ouro pela comunidade científica. Uma etapa importante na proposta de soluções para a automatização do MMSE é a classificação das imagens obtidas no teste do desenho do pentágono.

As redes neurais convolucionais mostraram um bom desempenho no geral com uma acurácia média de 93% na maior parte das arquiteturas descritas nesse trabalho. Analisando a Figura 38 e a Figura 39 é possível verificar que no geral o modelo possui um bom desempenho para a extração de características e classificação, é possível notar, analisando o valor dos respectivos erros da Figura 39 e da Figura 39, que a classificação é bem precisa, não só para os exemplos abaixo como para a maioria das imagens (ver Figura 36). Logo é possível confiar a atividade de classificação a esses modelos, porém com a supervisão de um especialista.

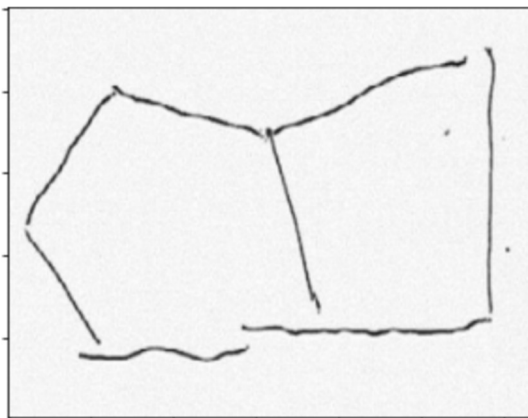


**Figura 38.** Imagem incorreta classificada corretamente pelo modelo com erro de 0,0070659.



**Figura 39.** Imagem correta classificada corretamente pelo modelo com erro de 0,00123647.

Uma análise mais minuciosa nos resultados demonstra que a rede embora com bons resultados não consegue estabelecer uma relação entre as características de forma segura, o resultado são imagens que contém traços e até um ou mais pentágonos, mas que não possuem a relação correta entre eles. Com isso é possível concluir que o modelo não está conseguindo extrair características suficientes para correlacionar a relação das duas figuras geométricas com a classe correta ou incorreta, um exemplo está na Figura 40 que contém uma imagem com apenas 1 pentágono e que foi classificada de forma incorreta.



**Figura 40.** Exemplo de imagem classificada incorretamente.

## 5.1. Trabalhos futuros

Esse trabalho tinha como objetivo propor uma arquitetura para a classificação das imagens do teste do pentágono, essa metodologia mostrou que possui um bom desempenho de classificação, mas não confiável na ausência de um especialista. Para alcançarmos níveis maiores de acurácia, é possível mudar a abordagem de classificação de imagens para localização de objetos em uma imagem e assim, ao invés de simplesmente classificar a imagem como um todo resolver o problema em duas etapas: a primeira localização de dois pentágonos e a segunda análise da relação entre os dois pentágonos localizados na imagem.

Essa segunda abordagem possui vantagem dupla, uma vez que trará um resultado mais assertivo para o caso exemplificado na Figura 37 e será possível aplicar a escala de Bourke para o desenho do pentágono ao invés de uma classificação binária como proposta no MMSE.

# Bibliografia

BOURKE, J.; CASTLEDEN C.M.; STEPHEN, R.; DENNIS, M.; **Short report: a comparison of clock and pentagon drawing in Alzheimer's disease**, 1995.

COLLIE, A.; MARUFF, P.; **An analysis of systems of classifying mild cognitive impairment in older people**, Aust. N. Z. J Psychiatry. 36 (2002) 133–140.

CORMARK, F.; AARSLAND, D.; BALLARD, C.; TOVÉE, M.J.; **Pentagon drawing and neuropsychological performance in Dementia with Lewy Bodies, Alzheimer's disease, Parkinson's disease and Parkinson's disease with dementia**, 2004.

ESCHWEILER, G.; LEYHE, T.; KLÖPPEL, S.; HÜLL, M.; **New developments in the diagnosis of dementia**, Dtsch Arztebl Int 107 (39) 2010 677-683.

FOLSTEIN, M.F.; FOLSTEIN, S.E.; MCHUGH, P.R.; **'Mini-mental state'. a practical method for grading the cognitive state of patients for the clinician**, J Psychiatr Res, 1975.

FOUNTOULAKIS et al.: **The standardised copy of pentagons test. Annals of General Psychiatry** 2011 10:13.

GOOGLE COLAB, Google Colab, 2018, Disponível em: <  
<http://colab.research.google.com/>>

GRAND, J.H.; CASPAR, S.; MACDONALD S.W.; **Clinical features and multidisciplinary approaches to dementia care**, J. Multidiscip. Health, 2011;

LECUN, Y.; BENGIO, Y.; HINTON, G.; **Deep learning**, Nature, Nature Research, v. 521, n. 7553, p. 436-444, 2015.

LECUN, Y.; BENGIO, Y.; **Convolutional networks for images, speech, and time series. The handbook of brain theory and neural networks**, 1995.

LI, Q.; CAI, W.; WANG, X.; ZHOU, Y.; FENG, D.D.; CHEN, M.; **Medical Image Classification with Convolutional Neural Network**, 2014 13th International Conference on Control, Automation, Robotics & Vision.



LINS, A.J.C.C.; MUNIZ M.T.C.; GARCIA, A.N.M.; GOMES, A.V.; CABRAL, R.M.; BASTOS-FILHO, C.J.A.; **Using artificial neural networks to select the parameters for the prognostic of mild cognitive impairment and dementia in elderly individuals**, 2017.

MARTINELLI, J. E.; CECATO, J.F.; MARTINELLI, M.O.; DE MELO, B.A.R.; APRAHAMIAN, I.; **Performance of the Pentagon Drawing test for the screening of older adults with Alzheimer's dementia**, 2018.

MIYAZAKI, C.K.; **Redes neurais convolucionais para aprendizagem e reconhecimento de objetos 3D**, 2017.

TOMBAUGH, T.N.; MCINTYRE, N.J.; **The mini-mental state examination: a comprehensive review**, J. Am. Geriatrics Soc., USA 40 (9) (1992) 922-935.

WORLD HEALTH ORGANIZATION (WHO). **Dementia: A public health priority**, Technical Report, 2012.