

UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS

FABIO AUGUSTO GONÇALVES FACCINI

Avaliação de desempenho de uma rede neural convolucional para
classificação de imagens ruidosas

São Carlos
2018

FABIO AUGUSTO GONÇALVES FACCINI

Avaliação de desempenho de uma rede neural convolucional para
classificação de imagens ruidosas

Monografia apresentada ao Curso de Engenharia Elétrica com ênfase em Eletrônica, da Escola de Engenharia de São Carlos da Universidade de São Paulo, como parte dos requisitos para obtenção do título de Engenheiro Eletricista/Eletrônico.

Orientador: Prof. Dr. Adilson Gonzaga

São Carlos

2018

AUTORIZO A REPRODUÇÃO TOTAL OU PARCIAL DESTE TRABALHO, POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

Ficha catalográfica elaborada pela Biblioteca Prof. Dr. Sérgio Rodrigues Fontes da EESC/USP com os dados inseridos pelo(a) autor(a).

F137a Faccini, Fábio Augusto Gonçalves
Avaliação de desempenho de uma rede neural convolucional para classificação de imagens ruidosas / Fábio Augusto Gonçalves Faccini; orientador Prof. Dr. Adilson Gonzaga. São Carlos, 2018.

Monografia (Graduação em Engenharia Elétrica com ênfase em Eletrônica) -- Escola de Engenharia de São Carlos da Universidade de São Paulo, 2018.

1. Redes neurais convolucionais. 2. Ruído. 3. Classificação de imagens. 4. Rede profunda. I. Título.

FOLHA DE APROVAÇÃO

Nome: Fábio Augusto Gonçalves Faccini

Título: "Avaliação de desempenho da acurácia de uma CNN para classificar imagens corrompidas por ruído"

Trabalho de Conclusão de Curso defendido e aprovado
em 19/11/2018,

com NOTA 10,0 (DEZ, ZERO), pela Comissão Julgadora:

Prof. Associado Adilson Gonzaga - Orientador - SEL/EESC/USP

Dra. Carolina Toledo Ferraz - Pós-Doutorado/Faculdade Campo Limpo Paulista

Mestre Adriane Cavichioli - Centro Estadual de Educação Tecnológica Paula Souza/Faculdade de Tecnologia de Presidente Prudente

Coordenador da CoC-Engenharia Elétrica - EESC/USP:
Prof. Associado Rogério Andrade Flauzino

AGRADECIMENTOS

Ao Prof. Dr. Adilson Gonzaga, pelo seu incentivo, dedicação e conhecimentos compartilhados que contribuíram para o desenvolvimento deste trabalho.

A minha família, pelo incentivo aos estudos durante toda minha vida.

A minha namorada, Monique, que sempre me apoiou nos momentos de dificuldades.

Aos colegas de faculdade, que foram companheiros durante todo o curso.

Agradeço a Universidade de São Paulo.

RESUMO

FACCINI, F. A. G. **Avaliação de desempenho de uma rede neural convolucional para classificação de imagens ruidosas**. 2018. 67 p. Monografia (Trabalho de Conclusão de Curso) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2018.

Este trabalho propõe a avaliação de desempenho de uma rede neural convolucional (CNN) no reconhecimento de imagens de dígitos numéricos corrompidos com ruído. Foi proposta uma rede profunda para classificação dos dígitos de 0 a 9 da base de dados MNIST, conjunto composto por imagens em escalas de cinza de dimensão 28x28, totalizando 60 mil imagens para treino e 10 mil imagens para teste. Foram gerados conjuntos de teste, a partir da base MNIST, corrompidos com ruídos do tipo gaussiano, impulsivo (sal e pimenta) e *speckle*. Usando a rede treinada com imagens sem ruído, sua robustez foi avaliada para os diferentes tipos de ruído, demonstrando uma drástica redução da acurácia para todos os casos. Em seguida, a rede foi treinada novamente, porém com adição desses três tipos de ruído no conjunto de treino separadamente. Os ruídos foram adicionados em diferentes proporções (quantidade de ruído e quantidade de imagens contaminadas) e para cada combinação, a rede foi treinada e testada. Além disso, para efeitos de comparação, a rede também foi avaliada com treinamento sem adição de ruído, porém utilizando-se técnicas convencionais de filtragem no conjunto de teste. As acurácias obtidas com a adição de ruído no treino aumentaram em até 80% para os ruídos gaussiano e impulsivo e 50% para o ruído *speckle*, superando o método de filtragem e se aproximando da capacidade de reconhecimento do ser humano. Isto é, apesar da CNN ser especializada em características de formatos devido sua natureza de filtragem, pode aprender informações estatísticas e aleatórias como as geradas por ruído.

Palavras chave: Redes neurais convolucionais. Ruído. Classificação de imagens. Rede profunda.

ABSTRACT

FACCINI, F. A. G. **Performance evaluation of a convolutional neural network for classification of noisy images**. 2018. 67 p. Monografia (Trabalho de Conclusão de Curso) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2018.

This study proposes the performance evaluation of a Convolutional Neural Network (CNN) for recognizing handwritten numerical digits images corrupted by noise. It was proposed a deep network for the classification of the 0 to 9 digits from the MNIST database, which is a set composed of 28x28 dimension grayscale images, totalizing 60 thousand training images and 10 thousand test images. Corrupt test sets were generated, from the MNIST database, with Gaussian, impulsive (salt and pepper) and speckle noise. By using the trained network with images without noise, its robustness was evaluated for the different types of noise, showing a drastically accuracy reduction for all the noise types. Then, the network was trained again, however adding to the training set these three types of noise separately. The noises were added in different proportions (amount of noise and amount of corrupted images) and for each combination, the network was trained and tested. In addition, for comparison purposes, the network was also evaluated with training without noise addition, but using conventional filtering techniques in the test set. The accuracies obtained with the addition of noise in the training have increased in up to 80% for the Gaussian and impulsive noise, and 50% for the speckle noise, overcoming the filtering method and reaching the human capacity of recognition. That is, notwithstanding the CNN be specialized in shape features due their filters nature, it can learn statistical and random information as those generated by noise.

Key words: Convolutional neural network. Noise. Image classification. Deep network.

LISTA DE ILUSTRAÇÕES

Figura 1 - Max Pooling.	26
Figura 2 - Imagem mostra a adição de ruído sal e pimenta. A esquerda tem-se a imagem original, e a direita tem-se a imagem com adição de 30% de ruído.	30
Figura 3 - Adição de ruído gaussiano com média 0. A esquerda tem-se a imagem original, e a direita tem-se a imagem com adição de ruído.	31
Figura 4 - Imagem mostra a adição de ruído speckle. A esquerda tem-se a imagem original, e a direita tem-se a imagem com adição de ruído.	31
Figura 5 - Arquitetura da CNN.	34
Figura 6: Imagens com adição de ruído de impulso (sal e pimenta). Acima e à esquerda está a imagem original (0% de ruído). Na sequência imagens corrompidas com 5%, 10%, 15%, 20%, 40%, 60%, 80%, 100% de pixels contaminados.	35
Figura 7: Imagens referentes a Figura 6 filtradas com filtro da mediana 3x3.	35
Figura 8: Imagens com adição de ruído Gaussiano. Acima e à esquerda está a imagem original. Na sequência imagens corrompidas com ruído Gaussiano com variância de 0,01; 0,025; 0,05; 0,075; 0,1; 0,3; 0,5 e 0,8.	36
Figura 9: Imagem referente a Figura 8 filtrada com filtro da média 3x3.	36
Figura 10: Imagens com adição de ruído speckle. Acima e à esquerda está a imagem original. Na sequência imagens corrompidas com ruído speckle com variância de 0,04; 0,1; 0,3; 0,7; 1,5; 5; 10 e 15.	37
Figura 11: Imagem referente a Figura 10 filtrada com filtro da média 3x3.	37
Figura 12: Desempenho da rede classificando imagens com ruído de impulso (CNN treinado com imagens sem ruído).	40
Figura 13: Desempenho da rede classificando imagens com ruído de impulso (CNN treinada com imagens com ruído em diferentes proporções).	42
Figura 14: Desempenho da rede classificando imagens com ruído de impulso (CNN treinada com diferentes quantidades de imagens contaminadas).	44
Figura 15: Comparação do desempenho da rede classificando imagens com ruído de impulso (Utilizando imagens no treino com e sem ruído, e com aplicação de filtros).	45
Figura 16: Desempenho da rede classificando imagens com ruído gaussiano (CNN treinada com imagens sem ruído).	47

Figura 17: Desempenho da rede classificando imagens com ruído gaussiano (CNN treinada com imagens com ruído em diferentes proporções).	49
Figura 18: Desempenho da rede classificando imagens com ruído gaussiano (CNN treinada com diferentes quantidades de imagens contaminadas).	51
Figura 19: Comparação do desempenho da rede classificando imagens com ruído gaussiano (Utilizando imagens no treino com e sem ruído, e com aplicação de filtros).	52
Figura 20: Desempenho da rede classificando imagens com ruído speckle (CNN treinada com imagens sem ruído).	54
Figura 21: Desempenho da rede classificando imagens com ruído speckle (CNN treinada com imagens com ruído em diferentes proporções).	55
Figura 22: Desempenho da rede classificando imagens com ruído speckle (CNN treinada com diferentes quantidades de imagens contaminadas).	57
Figura 23: Comparação do desempenho da rede classificando imagens com ruído speckle (Utilizando imagens no treino com e sem ruído, e com aplicação de filtros).	58

LISTA DE TABELAS

Tabela 1: Acurácias em Top-1 da CNN treinada com imagens sem ruído, para conjunto de teste de imagens com ruído de impulso (sal e pimenta).	39
Tabela 2: Acurácias da CNN em Top-1 para o conjunto de teste de imagens com ruído de impulso (sal e pimenta) e treinada com 10% das imagens do conjunto de treino contaminadas com o mesmo ruído, sendo 10% dos pixels dessas imagens contaminados.	41
Tabela 3: Acurácias da CNN em Top-1 para o conjunto de teste de imagens com ruído de impulso (sal e pimenta) e treinada com 10% das imagens do conjunto de treino contaminadas com o mesmo ruído, sendo 20% dos pixels dessas imagens contaminados.	41
Tabela 4: Acurácias da CNN em Top-1 para o conjunto de teste de imagens com ruído de impulso (sal e pimenta) e treinada com 10% das imagens do conjunto de treino contaminadas com o mesmo ruído, sendo 40% dos pixels dessas imagens contaminados.	41
Tabela 5: Acurácias da CNN em Top-1 para o conjunto de teste de imagens com ruído de impulso (sal e pimenta) e treinada com 10% das imagens do conjunto de treino contaminadas com o mesmo ruído, sendo 80% dos pixels dessas imagens contaminados.	41
Tabela 6: Acurácias da CNN em Top-1 para o conjunto de teste de imagens com ruído de impulso (sal e pimenta) e treinada com 30% das imagens do conjunto de treino contaminadas com o mesmo ruído, sendo 40% dos pixels dessas imagens contaminados.	43
Tabela 7: Acurácias da CNN em Top-1 para o conjunto de teste de imagens com ruído de impulso (sal e pimenta) e treinada com 60% das imagens do conjunto de treino contaminadas com o mesmo ruído, sendo 40% dos pixels dessas imagens contaminados.	43
Tabela 8: Acurácias da CNN em Top-1 para o conjunto de teste de imagens com ruído de impulso (sal e pimenta) e treinada com 90% das imagens do conjunto de treino contaminadas com o mesmo ruído, sendo 40% dos pixels dessas imagens contaminados.	43

Tabela 9: Acurácias em Top-1 da CNN treinada com imagens sem ruído, para conjunto de teste de imagens com ruído de impulso (sal e pimenta) tratadas com filtro da mediana 3x3.	45
Tabela 10: Acurácias em Top-1 da CNN treinada com imagens sem ruído, para conjunto de teste de imagens com ruído Gaussiano.	46
Tabela 11: Acurácias da CNN em Top-1 para conjunto de teste de imagens com ruído Gaussiano e treinada com 10% das imagens contaminadas com o mesmo ruído, utilizando variância igual a 0,075.	47
Tabela 12: Acurácias da CNN em Top-1 para conjunto de teste de imagens com ruído Gaussiano e treinada com 10% das imagens contaminadas com o mesmo ruído, utilizando variância igual a 0,1.	47
Tabela 13: Acurácias da CNN em Top-1 para conjunto de teste de imagens com ruído Gaussiano e treinada com 10% das imagens contaminadas com o mesmo ruído, utilizando variância igual a 0,3.	48
Tabela 14: Acurácias da CNN em Top-1 para conjunto de teste de imagens com ruído Gaussiano e treinada com 10% das imagens contaminadas com o mesmo ruído, utilizando variância igual a 0,8.	48
Tabela 15: Acurácias da CNN em Top-1 para conjunto de teste de imagens com ruído Gaussiano e treinada com 30% das imagens contaminadas com o mesmo ruído, utilizando variância igual a 0,3.	49
Tabela 16: Acurácias da CNN em Top-1 para conjunto de teste de imagens com ruído Gaussiano e treinada com 60% das imagens contaminadas com o mesmo ruído, utilizando variância igual a 0,3.	49
Tabela 17: Acurácias da CNN em Top-1 para conjunto de teste de imagens com ruído Gaussiano e treinada com 90% das imagens contaminadas com o mesmo ruído, utilizando variância igual a 0,3.	50
Tabela 18: Acurácias em Top-1 da CNN treinada com imagens sem ruído, para conjunto de teste de imagens com ruído de gaussiano tratadas com filtro da média 3x3.	51
Tabela 19: Acurácias em Top-1 da CNN treinada com imagens sem ruído, para conjunto de teste de imagens com ruído do tipo speckle.	53
Tabela 20: Acurácias da CNN em Top-1 para conjunto de teste de imagens com ruído speckle e treinada com 10% das imagens contaminadas com o mesmo ruído, utilizando variância igual a 0,7.	54

Tabela 21: Acurácias da CNN em Top-1 para conjunto de teste de imagens com ruído speckle e treinada com 10% das imagens contaminadas com o mesmo ruído, utilizando variância igual a 1,5.....	54
Tabela 22: Acurácias da CNN em Top-1 para conjunto de teste de imagens com ruído speckle e treinada com 10% das imagens contaminadas com o mesmo ruído, utilizando variância igual a 5.	55
Tabela 23: Acurácias da CNN em Top-1 para conjunto de teste de imagens com ruído speckle e treinada com 10% das imagens contaminadas com o mesmo ruído, utilizando variância igual a 15.	55
Tabela 24: Acurácias da CNN em Top-1 para conjunto de teste de imagens com ruído speckle e treinada com 30% das imagens contaminadas com o mesmo ruído, utilizando variância igual a 0,7.....	56
Tabela 25: Acurácias da CNN em Top-1 para conjunto de teste de imagens com ruído speckle e treinada com 60% das imagens contaminadas com o mesmo ruído, utilizando variância igual a 0,7.....	56
Tabela 26: Acurácias da CNN em Top-1 para conjunto de teste de imagens com ruído speckle e treinada com 90% das imagens contaminadas com o mesmo ruído, utilizando variância igual a 0,7.....	56
Tabela 27: Acurácias em Top-1 da CNN treinada com imagens sem ruído, para conjunto de teste de imagens com ruído de speckle tratadas com filtro da média 3x3.	57

LISTA DE ABREVIATURAS E SIGLAS

CNN – Convolutional Neural Network

ELU - Exponential Linear Units

FDP – Função Densidade de Probabilidade

ILSVRC – ImageNet Large Scale Visual Recognition Challenge

PCA - Principal Component Analysis

ReLU – Rectified Linear Unit

SUMÁRIO

1	INTRODUÇÃO	21
1.1	HISTÓRIA DA CNN	21
1.2	MOTIVAÇÃO	21
1.3	OBJETIVOS	23
1.4	ORGANIZAÇÃO DO TRABALHO	23
2	REVISÃO BIBLIOGRÁFICA	25
2.1	REDES NEURAS CONVOLUCIONAIS – ARQUITETURA	25
2.1.1	Entrada	25
2.1.2	Convolução	25
2.1.3	<i>Pooling</i>	26
2.1.4	Camada de não-linearidade	27
2.1.5	Camada totalmente conectada	27
2.1.6	Classificador Softmax.....	28
2.1.7	Normalização em lote.....	29
2.2	TREINAMENTO	29
2.3	TIPOS DE RUÍDOS	30
2.3.1	Ruído Impulsivo.....	30
2.3.2	Ruído Gaussiano	30
2.3.3	Ruído <i>Speckle</i>	31
2.4	MÉTODOS DE REMOÇÃO DE RUÍDO	32
2.4.1	Filtro da média.....	32
2.4.2	Filtro da mediana	32
3	MÉTODO DE AVALIAÇÃO PROPOSTO	33
3.1	SELEÇÃO DA BASE DE DADOS.....	33
3.2	ARQUITETURA DA REDE PROPOSTA	33
3.3	DETALHES DA APRENDIZAGEM	34
3.4	CONJUNTOS GERADOS PARA TREINO E TESTE	34
3.4.1	Ruído impulsivo	35
3.4.2	Ruído Gaussiano	35

3.4.3	Ruído <i>speckle</i>	36
3.5	ETAPAS DE TREINAMENTO E TESTES DA CNN.....	37
3.5.1	Experimento 1.....	37
3.5.2	Experimento 2.....	38
3.5.3	Experimento 3.....	38
3.5.4	Experimento 4.....	38
4	RESULTADOS	39
4.1	RUÍDO IMPULSIVO.....	39
4.1.1	Experimento 1.....	39
4.1.2	Experimento 2.....	40
4.1.3	Experimento 3.....	42
4.1.4	Experimento 4.....	44
4.2	RUÍDO GAUSSIANO.....	46
4.2.1	Experimento 1.....	46
4.2.2	Experimento 2.....	47
4.2.3	Experimento 3.....	49
4.2.4	Experimento 4.....	51
4.3	RUÍDO SPECKLE.....	53
4.3.1	Experimento 1.....	53
4.3.2	Experimento 2.....	54
4.3.3	Experimento 3.....	56
4.3.4	Experimento 4.....	57
5	CONCLUSÃO	59
	REFERÊNCIAS BIBLIOGRÁFICAS	61
	APÊNDICE A - ROTINA BASE PARA TREINAR E TESTAR CNN	65
	APÊNDICE B – CURVA GERADA PELO MATLAB DURANTE O TREINO	67

1 INTRODUÇÃO

1.1 HISTÓRIA DA CNN

As Redes Neurais Convolucionais, do inglês *Convolutional Neural Network* (CNN) pertencem a uma categoria específica de redes neurais com aprendizagem profunda que são capazes de aprender as características de imagens automaticamente, superando muitas técnicas convencionais de extração de características em visão computacional. Elas foram desenvolvidas a partir da análise dos processos biológicos no córtex visual dos animais para serem utilizadas na área de processamento de imagens. Foi observado que o córtex visual do gato continha células que eram acionados por pequenas regiões do campo visual chamadas de campo receptivo, a vizinhança dessas células possuía campos receptivos que se sobrepunham, e essas regiões em conjunto compunham o campo visual (HUBEL; WIESEL, 1968).

Essa premissa deu origem ao desenvolvimento das arquiteturas das CNN's, no qual a imagem passa por várias camadas de filtros, que atuam como as células no reconhecimento de características específicas, para no final ser capaz de reconhecer e categorizar uma imagem.

A primeira CNN desenvolvida foi a LeNet (LECUN et al., 1998), para a classificação de dígitos manuscritos. No entanto, somente a partir de 2012 que essas arquiteturas passaram a ter um enfoque em visão computacional, quando o AlexNet (KRIZHEVSKY; SUTSKEVER; HINTON, 2012) se tornou vencedora do ILSVRC (*ImageNet Large Scale Visual Recognition Challenge*), competição que avalia performance de algoritmos para reconhecimento de imagens, conseguindo uma taxa de erro em Top - 5 de 16,4%. A partir desse ano todos os vencedores do ILSVRC foram CNN's: GoogleLeNet (SZEGEDY et al., 2015) em 2014, VGG Net (SIMONYAN; ZISSERMAN, 2014) em 2014 (não foi a vencedora mas se tornou bastante conhecida) e ResNET (HE et al., 2016) em 2015, está última superando a capacidade de reconhecimento do ser humano, que gira em torno de 5% a 10% de erro, atingindo uma taxa de erro em Top – 5 de aproximadamente 3,6%.

1.2 MOTIVAÇÃO

Técnicas de aprendizagem profunda são apontadas como “estado da arte” em várias aplicações, como detecção e reconhecimento de objetos (SZEGEDY; TOSHEV; ERHAN, 2013), reconhecimento de faces (SCHROFF; KALENICHENKO; PHILBIN, 2015), sistemas de monitoramento (LEMLEY; CORCORAN, 2017), segmentação de imagens (LONG; SHELHAMER; DARRELL, 2015) e classificação (KRIZHEVSKY; SUTSKEVER; HINTON, 2012). Em tais abordagens baseadas em aprendizagem profunda, o sinal de entrada (imagem) é processado por unidades consecutivas de processamento de sinais que reorientam os dados de entrada para a forma mais representativa considerando as amostras alvo. As unidades de processamento de sinais são conhecidas como camadas que podem ser convolucionais ou totalmente conectadas. Essas camadas aplicam filtros (*kernels*) à sua entrada cujos parâmetros são aprendidos na etapa de treinamento. Uma CNN é composta por camadas alternadas de convolução onde uma camada tem o mesmo número de filtros e camadas diferentes podem ter quantidade de filtros diferentes, camadas de *downsampling*, que reduzem a quantidade de dados, e camadas totalmente conectadas que funcionam como classificador (KRIZHEVSKY; SUTSKEVER; HINTON, 2012). Para obter uma melhor convergência, várias outras técnicas incluindo *drop-out* (SRIVASTAVA et al., 2014) e normalização (IOFFE; SZEGEDY, 2015) são apresentadas na literatura. No entanto, apesar do alto desempenho conseguido, as CNN's possuem algumas restrições tais como: tamanho de imagem pré-definido e fixo para uma determinada topologia de rede, aprendizagem de características mais determinantes de forma e cor, baixa robustez à rotação da imagem e ao ruído.

Um dos principais problemas que levam à degradação da qualidade de imagens é o ruído, dados indesejáveis que causam diferenças na intensidade dos pixels. O ruído pode contaminar as imagens durante a transmissão, a recepção, ou pode ser causado por outros fatores como alta temperatura do sensor, baixo nível de luz, etc. (PANDEY; SINGH, 2015). Em visão computacional, um passo utilizado antes do processamento efetivo da imagem visando o reconhecimento é a redução do ruído por meio filtragem linear ou não-linear, no domínio do espaço ou da frequência. Esta abordagem, no entanto, tem o inconveniente de não preservar informações de bordas, dificultando a extração de características de formato dos objetos (KAUR; SINGH, 2015). Apesar do desempenho impressionante na classificação de objetos, as redes profundas demonstraram ser suscetíveis a amostras corrompidas com ruído (DODGE;

KARAM, 2016). Uma rede neural profunda identifica corretamente a imagem de entrada como “gato persa” com 99% de acurácia se nenhum ruído corromper a imagem. No entanto, quando uma pequena quantidade de ruído Gaussiano é adicionada, a rede classifica erroneamente a imagem como “chow-chow”. No reconhecimento facial o ruído mostrou ser determinante para o desempenho da rede, aumentando-se o nível de ruído afeta-se o desempenho da rede negativamente. Com um nível de ruído muito grande, a acurácia das redes previamente treinadas torna-se menor que 6%. Contudo, o olho humano ainda pode facilmente reconhecer uma face com este nível de contaminação por ruído. O desempenho de redes previamente treinadas como GoogLeNet e AlexNet cai mais rapidamente que o da rede VGG-Face (PARKHI et al., 2015) em relação ao aumento de ruído Gaussiano (KARAHAN et al., 2016). Alguns pesquisadores têm utilizado também as CNN’s para realizar a filtragem do ruído (XIE; XU; CHEN, 2012; ZHANG et al., 2017) ou associando a rede à Análise de Componentes Principais (PCA) para detectar qual é o tipo de ruído existente na imagem, visando posterior filtragem (KHAW et al., 2017).

1.3 OBJETIVOS

O objetivo deste trabalho é avaliar o desempenho de uma CNN quando imagens de teste são contaminadas com ruído, tendo sido a rede treinada com imagens sem ruído. Além disso, introduzir imagens ruidosas no conjunto de treinamento e observar qual o impacto na performance da rede. E por fim, também avaliar a CNN quando são aplicados filtros para remoção de ruído nas imagens de teste, sendo treinada apenas com imagens sem ruído.

1.4 ORGANIZAÇÃO DO TRABALHO

Este trabalho está dividido em 5 seções: na Seção I foi apresentado desde a história das CNN até a influência negativa do ruído no desempenho das Redes Neurais Convolucionais profundas; a Seção II apresenta a revisão bibliográfica necessária para o entendimento do trabalho; A Seção III apresenta o método proposto para avaliação de desempenho de uma CNN na classificação de dígitos numéricos manuscritos; os resultados obtidos são delineados na Seção VI e a Seção V conclui o trabalho.

2 REVISÃO BIBLIOGRÁFICA

2.1 REDES NEURAS CONVOLUCIONAIS – ARQUITETURA

A arquitetura básica da CNN é composta por uma estrutura formada por várias camadas. Sendo formada por uma camada de convolução, uma não-linearidade, uma camada de *downsampling* e por fim uma camada totalmente conectada. No entanto também existem outras camadas que podem ser incluídas para melhorar o treinamento e desempenho da rede. Abaixo estão descritas essas camadas presentes nas CNN's.

2.1.1 Entrada

As redes neurais possuem uma camada de entrada onde é inserida uma imagem que é interpretada computacionalmente como uma matriz de intensidades. Esta matriz pode ser uma matriz com profundidade 1, para o caso de imagens em tons de cinza, ou com profundidade 3, para o caso de serem imagens coloridas, onde cada canal RGB é representado por um nível de profundidade.

2.1.2 Convolução

Esta camada é responsável por aplicar filtros de convolução nas imagens de entrada. Durante o cálculo da convolução, são geradas matrizes de dimensão 2 denominadas mapas de ativação. Os filtros dessa camada são capazes de reconhecer certos padrões nas imagens atuando como *template matching*. Os filtros das camadas de convolução superiores reconhecem padrões mais simples, como bordas horizontais e verticais, enquanto as camadas convolucionais mais profundas reconhecem padrões mais complexos.

Cada filtro irá gerar um mapa de ativação e a junção de todos esses mapas será a saída da camada de convolução. Nesta camada, os hiperparâmetros (parâmetros que não são aprendidos durante o treinamento, e sim estipulados pelo desenvolvedor da CNN) a serem definidos são: tamanho do filtro (F), número de filtros (K), passo (S) e *padding* (P). A saída desta camada é uma matriz que possui dimensões ($W_2 \times H_2 \times D_2$), considerando uma matriz de entrada de dimensões ($W_1 \times H_1 \times D_1$), descrita pelas Equações 1, 2 e 3 (LI; KARPATY; JOHNSON, 2017a).

$$W_2 = \frac{(W_1 - F + 2P)}{s} + 1 \quad (1)$$

$$H_2 = \frac{(H_1 - F + 2P)}{s} + 1 \quad (2)$$

$$D_2 = K \quad (3)$$

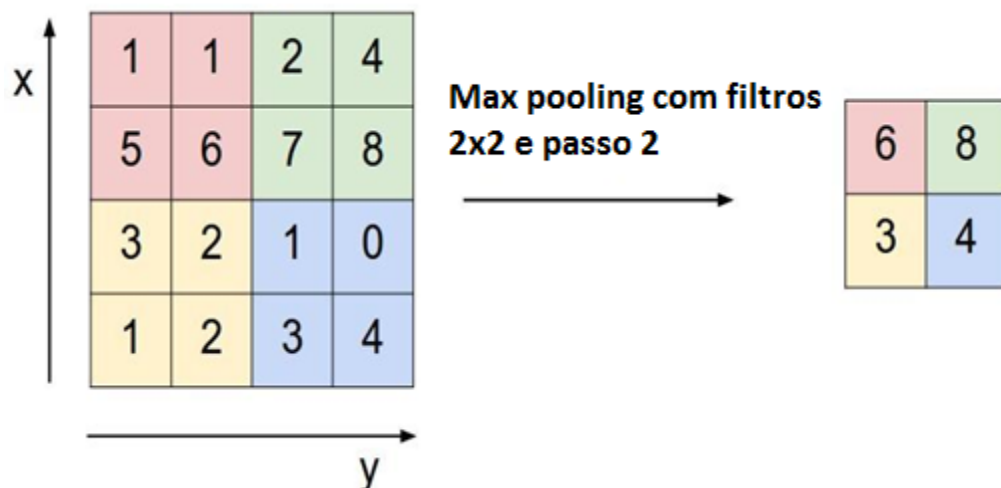
2.1.3 Pooling

A camada de *pooling* possui como principal função tornar a rede invariante no espaço, reduzindo o tamanho das imagens (*downsampling*). Ela também torna a rede robusta a sobreajuste, além de acelerar os cálculos matemáticos envolvidos nas camadas mais profundas ao reduzir a quantidade de dados a serem processados.

Existem várias funções que realizam a operação de *pooling*, as mais utilizadas são *Average pooling* e *MAX pooling*, esta última demonstrando ser superior na detecção de invariâncias quando comparada a operação de *subsampling* (SCHERER; MÜLLER; BEHNKE, 2010) e sendo a mais comum em CNN's.

A função *MAX pooling* aplica uma função janela em cada mapa de ativação, computando apenas o valor máximo de cada vizinhança, reduzindo a matriz de entrada em comprimento e largura, mas mantendo a mesma profundidade. A Figura 1 exemplifica esta operação.

Figura 1 - Max Pooling.



Vale ressaltar que a operação de *MAX pooling*, ao manter os valores mais altos, mantêm os valores em que os filtros das camadas anteriores foram mais acionados, preservando as informações importantes que a camada de convolução extraiu.

Para as CNN's, a camada de *pooling* necessita de dois hiperparâmetros: passo (S) e extensão espacial do filtro (F). Considerando sua entrada como uma matriz de extensão $(W_1 \times H_1 \times D_1)$, sua saída é uma matriz de dimensão $(W_2 \times H_2 \times D_2)$ determinada pelas Equações 4, 5 e 6 (LI; KARPATY; JOHNSON, 2017a).

$$W_2 = \frac{(W_1 - F)}{s} + 1 \quad (4)$$

$$H_2 = \frac{(H_1 - F)}{s} + 1 \quad (5)$$

$$D_2 = D_1 \quad (6)$$

2.1.4 Camada de não-linearidade

Esta camada tem como objetivo decidir se cada região da imagem foi acionada ou não, simulando o acionamento de um neurônio. A função mais usada em redes neurais convolucionais é a função denominada como *Rectified Linear Units* (ReLU) (NAIR; HINTON; 2010). No entanto, existem outras funções, como tanh e sigmoid, que também podem ser utilizadas, além de derivações da ReLU, como por exemplo *Leaky ReLU* e *Exponential Linear Units* (ELU). Contudo, redes que utilizam ReLU e suas derivadas usualmente convergem mais rápido do que quando treinadas com outras funções de ativação (KRIZHEVSKY; SUTSKEVER; HINTON, 2012).

A função ReLU é definida pela Equação 7, onde todos os valores abaixo de 0 são considerados 0 e os valores acima seguem uma função linear. Essa camada não possui hiperparâmetros.

$$f(x) = \max(0, x) \quad (7)$$

2.1.5 Camada totalmente conectada

Ela é a última camada da arquitetura, recebe todos os elementos da camada anterior e dispõe eles como um vetor. Assim, cada pixel oriundo da camada anterior corresponde a um neurônio de entrada na camada totalmente conectada, e a

quantidade de neurônios de saída corresponde ao número de elementos para a classificação, sendo que cada elemento ao final do processo computacional terá um *score*. Ela recebe esse nome pois, dentro dela, cada neurônio de uma camada interna é conectado a todos os outros neurônios da camada anterior.

A camada totalmente conectada processa as informações da camada anterior, que representa os mapas de ativação de características de alto nível, e determina quais características se relacionam mais com determinada classe, assim gera *scores* para cada uma delas. Por exemplo, se a rede está prevendo que uma imagem é um avião, a entrada da camada totalmente conectada terá mapas de ativação que representam características de aviões, e assim, o neurônio de saída que representa o elemento avião terá um *score* alto, enquanto os outros neurônios não. Isso se aplica a todos os elementos que a rede prevê.

2.1.6 Classificador Softmax

O classificador *Softmax* recebe esse nome pois faz uso da função *Softmax*, que normaliza os *scores* das classes recebidas da camada anterior em valores que totalizam 1, demonstrada pela Equação 8.

$$f_j(z) = \frac{e^{z_j}}{\sum_k e^{z_k}} \quad (8)$$

Onde z_j simboliza o elemento de número j do vetor de *scores*. O denominador da Equação 8 representa o somatório de e elevado aos *scores*, enquanto o numerador representa e elevado a cada *score*.

O classificador *Softmax* aplica uma função erro denominada *cross entropy loss*, exibida pela Equação 9.

$$L_i = -\log \left(\frac{e^{f_{yi}}}{\sum_j e^{f_{yj}}} \right) \quad (9)$$

A Equação 9 calcula a probabilidade log dos valores normalizados (a parte dentro da equação representa a função *Softmax*) (LI; KARPATY; JOHNSON, 2017b). O valor do erro L_i é utilizado durante o treinamento da rede. A função assume

que cada elemento pertence a somente uma classe, portanto, se existem elementos que pertençam a múltiplas classes o classificador *Softmax* não é indicado.

2.1.7 Normalização em lote

Esta camada tem o intuito de acelerar e facilitar o processo de convergência da CNN ao reduzir o *Internal Covariate Shift*, fenômeno que ocorre devido ao fato da distribuição estatística da entrada de cada camada mudar conforme os parâmetros da rede são atualizados durante o treino, incluindo camadas de normalização antes das entradas das demais camadas.

Este método acelera o processo de treinamento das redes profundas, torna possível o uso de taxas de aprendizagem maiores e torna a inicialização dos parâmetros da rede menos crítica (IOFFE; SZEGEDY, 2015).

2.2 TREINAMENTO

As camadas descritas acima são os processos utilizados na CNN que fazem a predição das classes para uma determinada imagem de entrada. No entanto, os parâmetros internos da rede precisam ser previamente treinados para alcançar uma alta acurácia.

O treinamento da CNN é o processo matemático pelo qual a rede irá atualizar seus parâmetros tendendo a minimizar o erro. Ele consiste da utilização conjunta das técnicas de *backpropagation*, que demonstrou ser um método eficaz para a atualização dos parâmetros da rede (LECUN et al., 1989) e gradiente descendente. Primeiramente, a imagem de entrada passa por todas as camadas descritas anteriormente (*forward propagation*), saindo como um vetor de *scores* que é utilizado para o cálculo de um erro. Então, é feito o cálculo dos gradientes da função erro no sentido do final da rede, para o início dela (*backpropagation*). Os valores dos gradientes calculados e os parâmetros obtidos no *forward propagation* são utilizados para atualizar os pesos de todas as camadas da rede.

Nesta etapa existe um importante hiperparâmetro: taxa de aprendizagem, fator que indica a escala das atualizações dos pesos da rede no sentido de minimizar o erro. Quanto maior, mais rápido a rede treina, porém a chance de não convergir aumenta; quanto menor, mais lento fica, porém a chance de convergência cresce.

2.3 TIPOS DE RUÍDOS

Nesta seção serão descritos os ruídos gaussiano, impulsivo (sal e pimenta) e *speckle* (VERMA; ALI, 2013) além de técnicas de filtragem comuns usadas para a remoção deles.

2.3.1 Ruído Impulsivo

O ruído de impulso também é conhecido como ruído sal e pimenta, ruído aleatório ou independente. Pontos pretos e brancos aparecem na imagem, dando origem ao nome “sal e pimenta”. Esse ruído surge na imagem devido a mudanças bruscas e repentinas do sinal da imagem. Problemas na fonte de aquisição de imagem ou componentes defeituosos superaquecidos podem causá-lo. A Figura 2 denota uma imagem com inserção de ruído sal e pimenta.

Figura 2 - Imagem mostra a adição de ruído sal e pimenta. A esquerda tem-se a imagem original, e a direita tem-se a imagem com adição de 30% de ruído.



Fonte: VERMA; ALI, 2013, p. 618.

2.3.2 Ruído Gaussiano

Também conhecido como erro normal, o ruído Gaussiano é um ruído estatístico do tipo aditivo, tal que cada pixel da imagem ruidosa representa a soma de seu valor original com um valor aleatório descrito por uma distribuição Gaussiana de densidade de probabilidade ou somente função densidade de probabilidade (FDP) $P(x)$ dada pela Equação 10, na qual σ é o desvio padrão, μ é a média e x é o valor da intensidade do pixel.

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (10)$$

As principais fontes de ruído Gaussiano em imagens digitais surgem durante a aquisição, como por exemplo, ruído no sensor causado por iluminação fraca e/ou alta temperatura e também durante a transmissão, gerado pelo circuito eletrônico. A Figura 3 demonstra a adição de ruído gaussiano em uma imagem.

Figura 3 - Adição de ruído gaussiano com média 0. A esquerda tem-se a imagem original, e a direita tem-se a imagem com adição de ruído.



Fonte: VERMA; ALI, 2013, p. 618.

2.3.3 Ruído *Speckle*

O ruído do tipo *speckle* é um ruído do tipo multiplicativo definido pela Equação 11, Sendo I a imagem e n um ruído aleatório uniformemente distribuído com média 0 e variância v .

$$J = I + n * I \quad (11)$$

Ruídos desse tipo surgem durante a aquisição de imagens, devido ao efeito das condições ambientais nos sensores. A Figura 4 demonstra uma imagem com adição de ruído *speckle*.

Figura 4 - Imagem mostra a adição de ruído *speckle*. A esquerda tem-se a imagem original, e a direita tem-se a imagem com adição de ruído.



Fonte: VERMA; ALI, 2013, p. 618 – 619.

2.4 MÉTODOS DE REMOÇÃO DE RUÍDO

2.4.1 Filtro da média

Filtro da média é um filtro linear que tem como característica principal suavizar a imagem ao reduzir a variação de intensidade entre pixels vizinhos. Ele calcula a média dos pixels vizinhos e substitui no pixel central. A Equação 12 exemplifica um núcleo de um filtro da média.

$$A = \begin{bmatrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{bmatrix} \quad (12)$$

Durante a convolução, neste caso, todos os nove pixels de cada passo serão multiplicados por 1/9 e somados, caracterizando o cálculo da média. Filtros da média são eficazes para ruído do tipo gaussiano e *speckle* (VERMA; ALI, 2013).

2.4.2 Filtro da mediana

O filtro da mediana é um filtro não linear, onde o pixel central é substituído pela mediana dos valores contidos dentro da região do filtro.

O filtro da mediana é efetivo para remoção de ruídos do tipo sal e pimenta (VERMA; ALI, 2013).

3 MÉTODO DE AVALIAÇÃO PROPOSTO

Para o desenvolvimento deste trabalho foi utilizado o software MATLAB R2018a juntamente com a Toolbox de Redes Neurais (*Neural Network Toolbox*) utilizando a CPU de um computador I5 com 4 Gb de RAM.

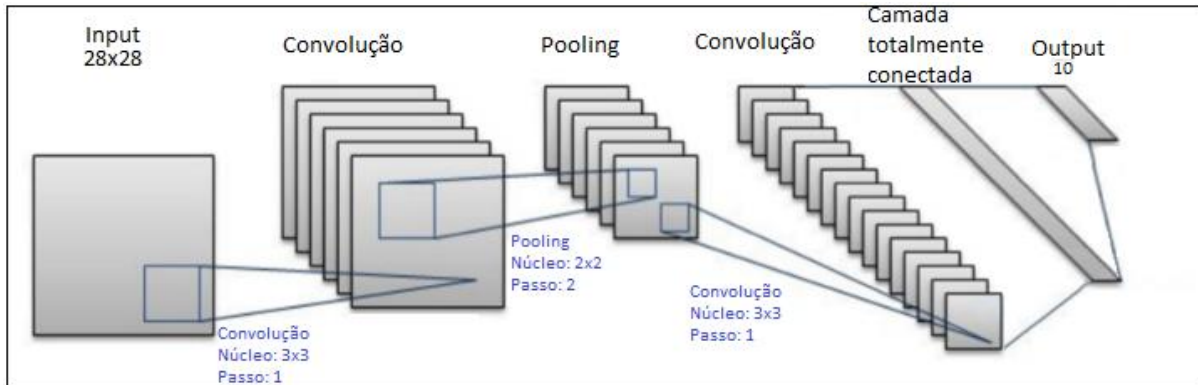
3.1 SELEÇÃO DA BASE DE DADOS

Para a avaliação de desempenho da acurácia da CNN proposta neste trabalho, foi utilizada a base de dados MNIST (LECUN; CORTES; BURGES, 1998), que é formada por um conjunto de dígitos manuscritos de 0 a 9 perfazendo um total de 60 mil imagens para treinamento e 10 mil imagens para teste. O reconhecimento de dígitos manuscritos é um problema importante em visão computacional, e tem sido usado em avaliações de métodos de reconhecimento de padrões e algoritmos de aprendizado de máquina há muitos anos. Todos os dígitos foram digitalizados em níveis de cinza, com tamanho normalizado e centralizado tal que o centro de gravidade de cada dígito esteja no centro da imagem com 28 x 28 pixels.

3.2 ARQUITETURA DA REDE PROPOSTA

As imagens da base de dados foram normalizadas para que ficassem com média zero e variância 1. A Figura 5 mostra a arquitetura da CNN proposta, contendo duas camadas de convolução, uma camada de *pooling* e uma camada totalmente conectada. A primeira camada da rede realiza a convolução com 25 filtros de tamanho 3x3, passo 1 e *padding* com zeros. As camadas seguintes são de normalização em lote, seguida da função de ativação ReLU. A camada de *pooling* executa a função *MAX pooling*, utilizando filtros 2x2 com passo 2, de modo que não haja sobreposição de pixels durante o processamento. A camada seguinte realiza a convolução com 32 filtros de tamanho 3x3 com passo 1 e *padding* com zeros. Na sequência foi utilizada outra camada de normalização em lote seguida da função de ativação ReLU. Finalmente foi definida uma camada totalmente conectada, com dez saídas (dígitos de 0 a 9). Os *scores* gerados foram normalizados utilizando-se a função *softmax*.

Figura 5 - Arquitetura da CNN.



Fonte: Adaptado de Sambit Mahapatra¹.

3.3 DETALHES DA APRENDIZAGEM

As imagens foram treinadas utilizando-se gradiente descendente estocástico com *momentum* e *batch size* de 128 amostras. Foi utilizada uma taxa de aprendizagem de 0,01.

Dentre as 60 mil imagens do conjunto de treino da base de dados MNIST, 15 mil foram separadas para validação, sobrando 45 mil para o treino.

3.4 CONJUNTOS GERADOS PARA TREINO E TESTE

Para a realização dos experimentos a base MNIST foi modificada adicionando-se ruídos do tipo gaussiano, impulsivo e *speckle*. Para fazer essa adição nas imagens foi utilizado o *app ImageBatchProcessor* do MATLAB, que permite selecionar uma grande quantidade de imagens e processá-las de maneira simples por meio de uma função criada pelo usuário. Para cada tipo citado anteriormente foram utilizadas oito proporções específicas de ruído, de modo que sua intensidade aumente gradativamente até a imagem deixar de ser facilmente reconhecida pelo olho humano. Essas proporções foram usadas para gerar conjuntos de treino e teste com ruído.

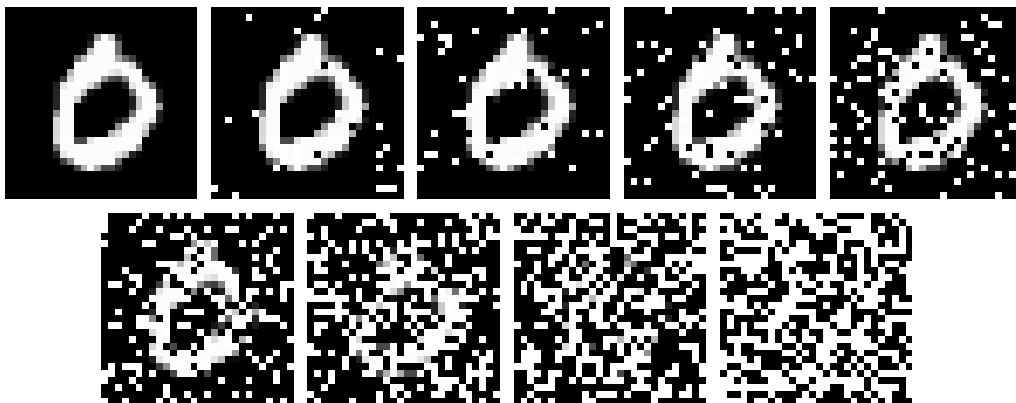
Além disso, as imagens geradas também foram tratadas com filtros específicos para verificar o desempenho da rede quando são utilizadas técnicas convencionais para remover ruído.

¹ Disponível em: <https://towardsdatascience.com/a-simple-2d-cnn-for-mnist-digit-recognition-a998dbc1e79a>; Acesso set. 2018.

3.4.1 Ruído impulsivo

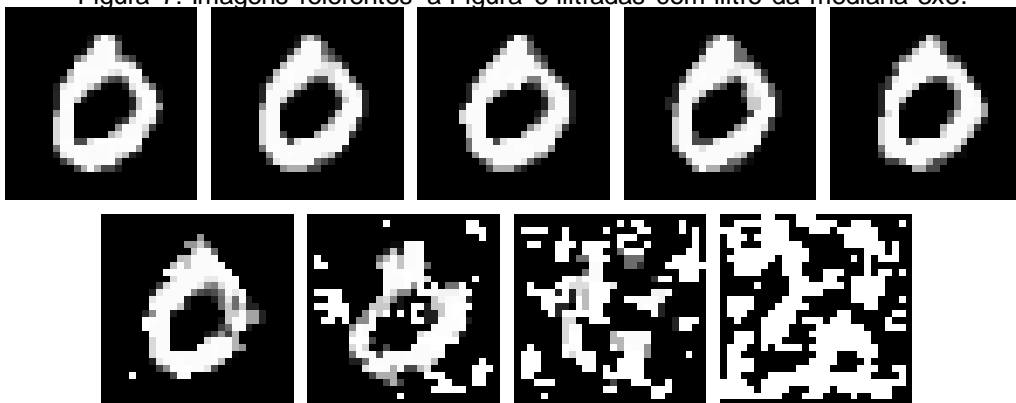
Para o ruído impulsivo foram escolhidas as proporções de 5%, 10%, 15%, 20%, 40%, 60%, 80%, 100% de pixels contaminados na imagem. A Figura 6 mostra a imagem de um dígito manuscrito “zero” do conjunto de teste da base utilizada, degradada com a adição de ruído de impulso. As imagens contaminadas com ruído sal e pimenta foram tratadas com filtro da mediana 3x3, as imagens estão demonstradas na Figura 7.

Figura 6: Imagens com adição de ruído de impulso (sal e pimenta). Acima e à esquerda está a imagem original (0% de ruído). Na sequência imagens corrompidas com 5%, 10%, 15%, 20%, 40%, 60%, 80%, 100% de pixels contaminados.



Fonte: Elaborada pelo autor.

Figura 7: Imagens referentes a Figura 6 filtradas com filtro da mediana 3x3.



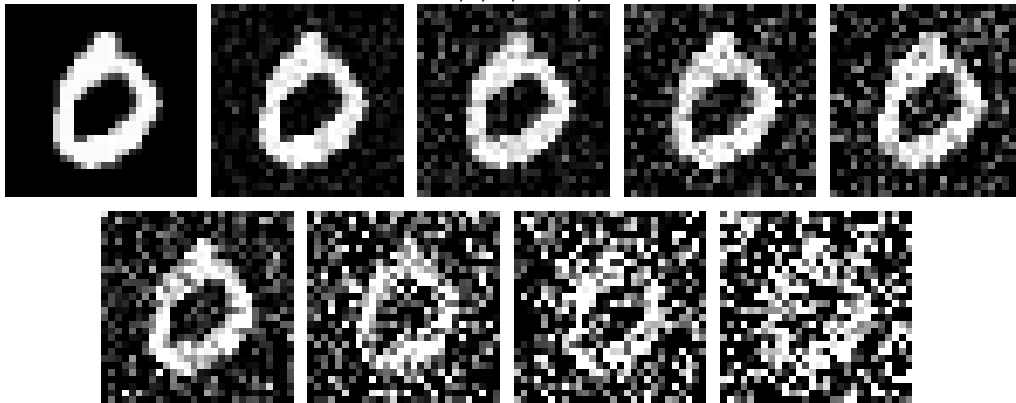
Fonte: Elaborada pelo autor.

3.4.2 Ruído Gaussiano

Para o ruído gaussiano foram escolhidos valores de variância igual a 0,01; 0,025; 0,05; 0,075; 0,1; 0,3; 0,5 e 0,8, utilizando média 0. A Figura 8 mostra um exemplo de

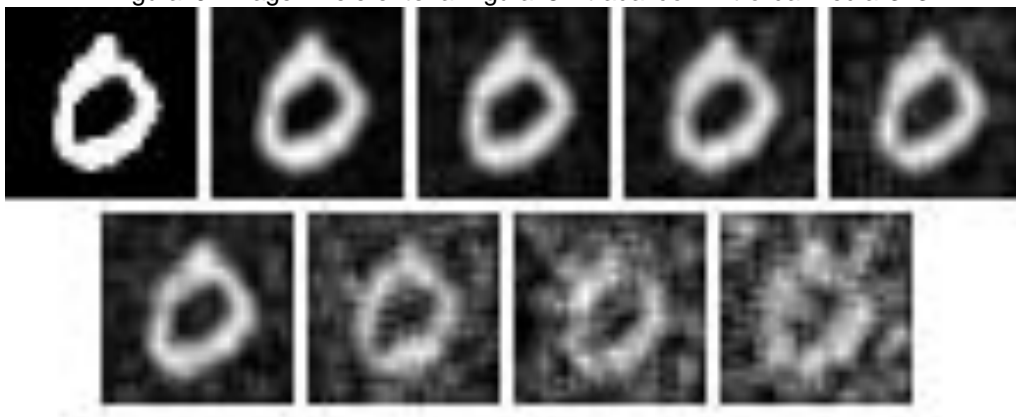
imagens do dígito manuscrito “zero” do conjunto de teste, degradadas com a adição de ruído gaussiano. As imagens contaminadas com ruído gaussiano foram tratadas com filtro da média 3x3. As imagens geradas estão demonstradas na Figura 9.

Figura 8: Imagens com adição de ruído Gaussiano. Acima e à esquerda está a imagem original. Na sequência imagens corrompidas com ruído Gaussiano com variância de 0,01; 0,025; 0,05; 0,075; 0,1; 0,3; 0,5 e 0,8.



Fonte: Elaborada pelo autor.

Figura 9: Imagem referente a Figura 8 filtrada com filtro da média 3x3.

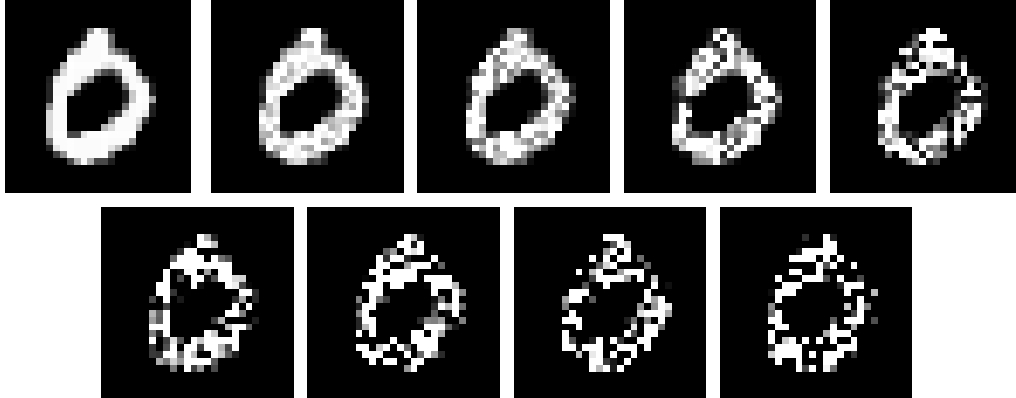


Fonte: Elaborada pelo autor.

3.4.3 Ruído *speckle*

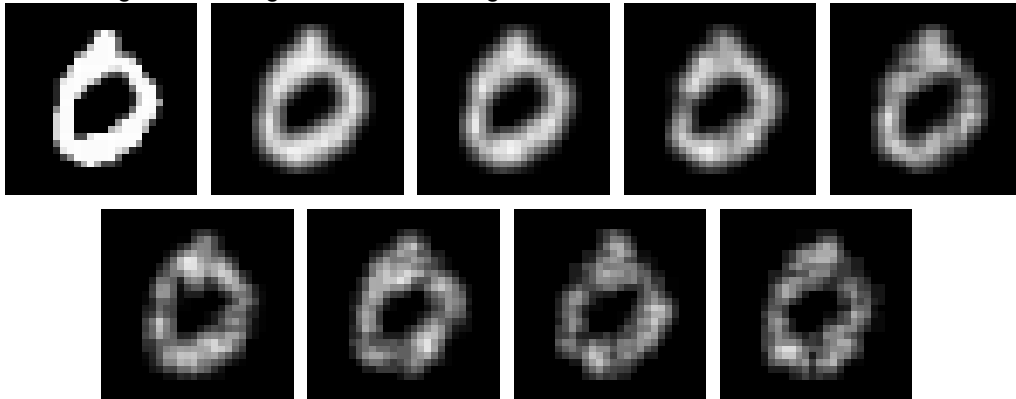
Para o ruído *speckle* foram escolhidos valores de variância igual a 0,04; 0,1; 0,3; 0,7; 1,5; 5; 10 e 15, com média 0. A Figura 10 mostra um exemplo de imagens do dígito manuscrito “zero” do conjunto de teste, degradadas com a adição de ruído *speckle*. As imagens contaminadas com ruído *speckle* foram tratadas com filtro da média 3x3 (Figura 11).

Figura 10: Imagens com adição de ruído *speckle*. Acima e à esquerda está a imagem original. Na sequência imagens corrompidas com ruído *speckle* com variância de 0,04; 0,1; 0,3; 0,7; 1,5; 5; 10 e 15.



Fonte Elaborada pelo autor.

Figura 11: Imagem referente a Figura 10 filtrada com filtro da média 3x3.



Fonte: Elaborada pelo autor.

3.5 ETAPAS DE TREINAMENTO E TESTES DA CNN

Foram feitas análises e testes para os três tipos de ruídos apresentados anteriormente. Para cada tipo foram preparados oito conjunto de testes, nas respectivas proporções apresentadas. Foram feitos 4 experimentos para cada tipo de ruído a fim de analisar diferentes aspectos e respostas da CNN.

3.5.1 Experimento 1

A rede foi treinada com a base de treino MNIST e foi testada pelos conjuntos de testes MNIST modificados com adição de ruído nas diferentes proporções previamente estabelecidas. Com isso, totalizaram-se nove conjuntos de testes (para

cada tipo de ruído): o conjunto de teste MNIST original mais oito conjuntos gerados com ruído, resultando em nove testes para uma única rede treinada.

3.5.2 Experimento 2

O conjunto de treino foi modificado com a adição de ruído em 10% de suas imagens em quatro proporções de ruído diferentes, totalizando quatro conjuntos de treino diferentes.

Para este caso, a rede precisou ser treinada uma vez para cada conjunto de treino e cada rede treinada foi testada pelos mesmos nove conjuntos de teste do Experimento 1.

3.5.3 Experimento 3

Para o Experimento 3 o conjunto de treino foi mais uma vez modificado. Nesse caso fixou-se uma proporção de ruído, variando a quantidade de imagens no conjunto de treino em que o ruído era adicionado. Assim, foram gerados conjuntos de treino inserindo ruído em 30%, 60% e 90% de suas imagens, totalizando três conjuntos de treino.

Novamente, a rede precisou ser treinada uma vez para cada conjunto e cada rede treinada foi testada pelos nove conjuntos de teste do Experimento 1.

3.5.4 Experimento 4

O último experimento, similar ao Experimento 1, consiste em treinar a rede com a base de treino MNIST original. Porém, utilizar os conjuntos de teste gerados a partir da utilização de filtros para remoção de ruído.

4 RESULTADOS

Os resultados serão divididos em 3 seções, em cada seção serão apresentados os quatro experimentos correspondentes a cada tipo de ruído. Vale ressaltar que todas as acurácias obtidas para os conjuntos de testes foram obtidas em Top – 1.

4.1 RUÍDO IMPULSIVO

4.1.1 Experimento 1

Abaixo está a Tabela 1 com as acurácias obtidas para a rede treinada com o conjunto de treino MNIST e testada com o conjunto de teste MNIST com diferentes proporções de ruído impulsivo inseridos.

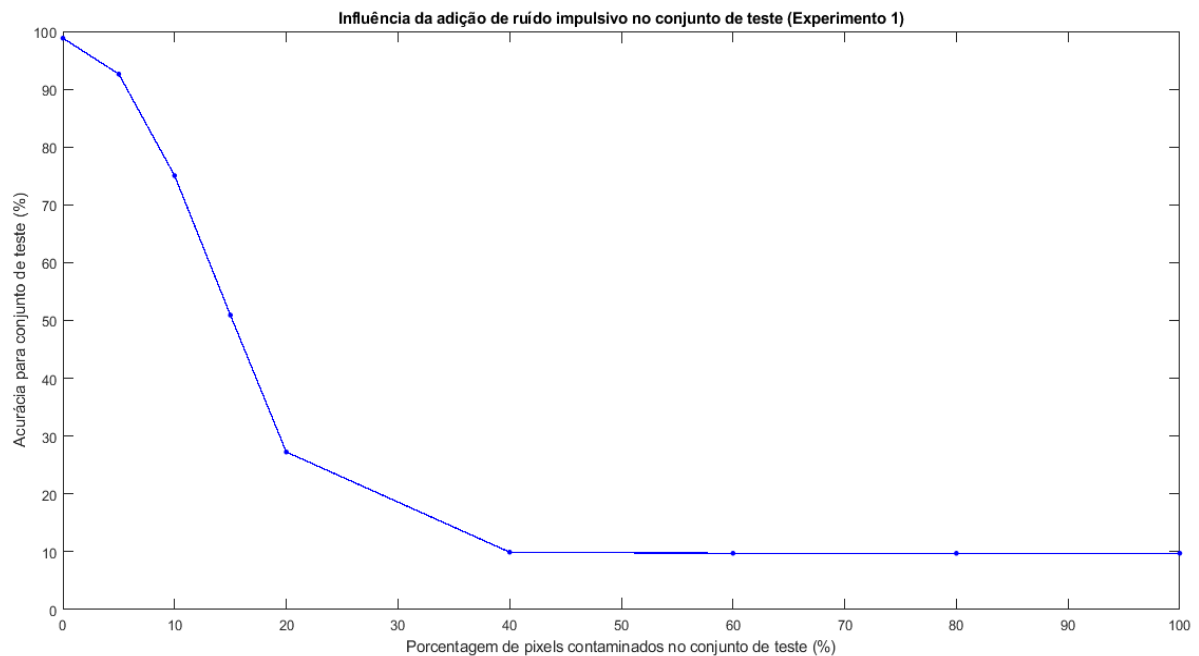
Tabela 1: Acurácias em Top-1 da CNN treinada com imagens sem ruído, para conjunto de teste de imagens com ruído de impulso (sal e pimenta).

% pixels contaminados	0	5	10	15	20	40	60	80	100
Acurácia	98,83	92,62	75,05	50,62	27,26	9,93	9,74	9,74	0,74

Fonte: Elaborada pelo autor.

O gráfico da Figura 12 mostra o desempenho da CNN treinada com imagens sem contaminação por ruído. Com o modelo gerado classifica-se as imagens de teste que são contaminadas por ruído de impulso.

Figura 12: Desempenho da rede classificando imagens com ruído de impulso (CNN treinado com imagens sem ruído).



Fonte: Elaborada pelo autor.

Pode-se observar que existe uma queda brusca no desempenho da rede, que a partir de 15% dos pixels contaminados, não opera mais como um classificador eficiente. No entanto, quando comparada com o olho humano, pode-se ainda reconhecer o dígito correspondente a quarta imagem da linha superior da Figura 6, que corresponde a uma degradação da amostra com ruído de impulso de 15%. Para seres humanos, apenas as 4 últimas imagens da linha inferior da Figura 6 não permitem uma identificação segura (ruído de impulso superior a 40%).

4.1.2 Experimento 2

Para o segundo experimento, observando-se as acurácias listadas na Tabela 1 e o desempenho da rede no gráfico da Figura 12, decidiu-se por contaminar 10% das imagens do conjunto de treinamento (ponto da Tabela 1 em que o classificador perde seu poder de discriminação, *random guess*, acertando cerca de 75% de classificações em Top-1) com quatro valores de proporção de ruído de impulso, mantendo os outros 90% do conjunto de treino sem alterações. Foram escolhidos os valores de contaminação igual a 10%, 20%, 40% e 80% de pixels contaminados. Os resultados obtidos são mostrados nas Tabelas 2, 3, 4 e 5 respectivamente.

Tabela 2: Acurácias da CNN em Top-1 para o conjunto de teste de imagens com ruído de impulso (sal e pimenta) e treinada com 10% das imagens do conjunto de treino contaminadas com o mesmo ruído, sendo 10% dos pixels dessas imagens contaminados.

% pixels contaminados	0	5	10	15	20	40	60	80	100
Acurácia	98,62	98,05	97,51	95,62	93,46	66,61	30,80	14,32	9,79

Fonte: Elaborada pelo autor.

Tabela 3: Acurácias da CNN em Top-1 para o conjunto de teste de imagens com ruído de impulso (sal e pimenta) e treinada com 10% das imagens do conjunto de treino contaminadas com o mesmo ruído, sendo 20% dos pixels dessas imagens contaminados.

% pixels contaminados	0	5	10	15	20	40	60	80	100
Acurácia	98,79	98,53	98,05	96,91	95,88	79,68	44,38	19,38	9,96

Fonte: Elaborada pelo autor.

Tabela 4: Acurácias da CNN em Top-1 para o conjunto de teste de imagens com ruído de impulso (sal e pimenta) e treinada com 10% das imagens do conjunto de treino contaminadas com o mesmo ruído, sendo 40% dos pixels dessas imagens contaminados.

% pixels contaminados	0	5	10	15	20	40	60	80	100
Acurácia	98,48	98,18	97,89	97,25	96,56	89,18	62,95	26,40	9,83

Fonte: Elaborada pelo autor.

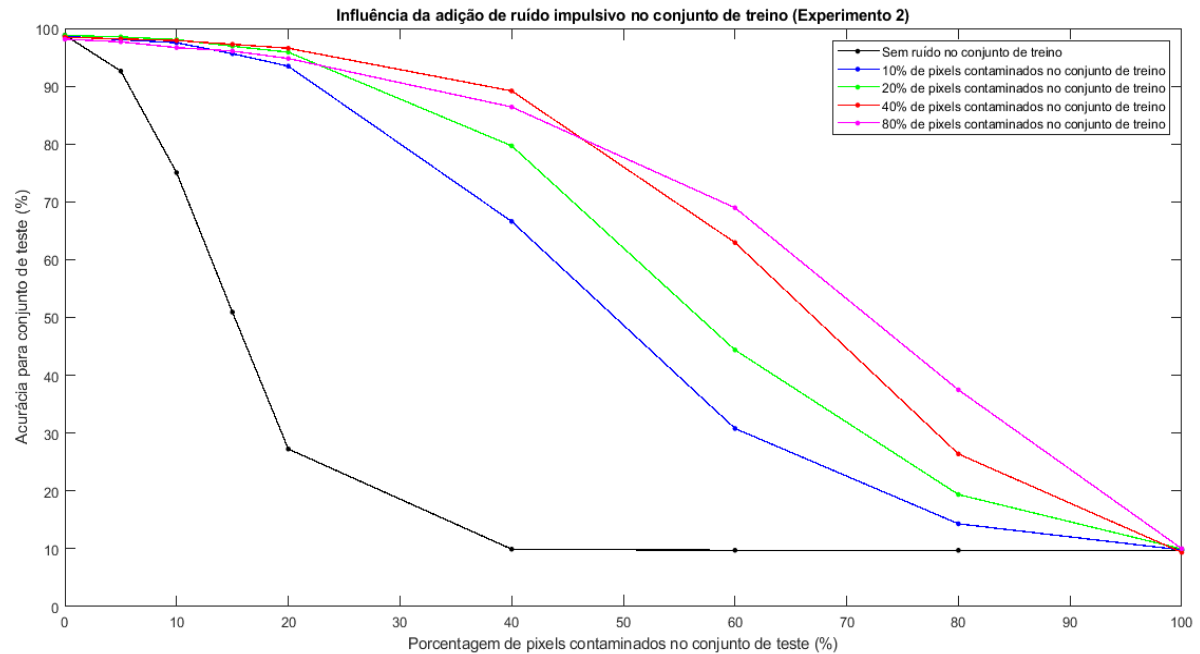
Tabela 5: Acurácias da CNN em Top-1 para o conjunto de teste de imagens com ruído de impulso (sal e pimenta) e treinada com 10% das imagens do conjunto de treino contaminadas com o mesmo ruído, sendo 80% dos pixels dessas imagens contaminados.

% pixels contaminados	0	5	10	15	20	40	60	80	100
Acurácia	98,13	97,67	96,97	96,08	94,79	86,40	68,97	37,49	10,02

Fonte: Elaborada pelo autor.

Para facilitar a disposição dos resultados, a Figura 13 evidencia as acurácias da CNN para os valores de contaminação no conjunto de treino demonstrados nas tabelas acima, juntamente com as acurácias encontradas no Experimento 1 (Tabela 1), onde não foi inserido ruído no conjunto de treino.

Figura 13: Desempenho da rede classificando imagens com ruído de impulso (CNN treinada com imagens com ruído em diferentes proporções).



Fonte: Elaborada pelo autor.

Pode-se observar que a rede aprendeu proporcionalmente o ruído introduzido melhorando o desempenho de classificação para as diferentes quantidades de ruído introduzidas. Os testes demonstraram que houve um acréscimo de até 79,25% com 40% dos pixels contaminados no conjunto de teste, para o caso de 40% de contaminação no conjunto de treino.

Constata-se que para todos os casos analisados a rede torna-se um classificador extremamente eficiente para até 20% dos pixels contaminados no conjunto de teste, e para os dois últimos casos com 40% dos pixels contaminados ainda pode ser considerado um bom classificador.

Foram atingidos valores mais altos de acurácia com a rede treinada com 40% de ruído no conjunto de treino para conjuntos de teste de até 40% de ruído, a partir desse ponto a rede treinada com 80% de ruído demonstrou uma acurácia maior.

4.1.3 Experimento 3

Para o Experimento 3 foi fixado o valor de 40% dos pixels contaminados no conjunto de treino e variado a quantidade de imagens contaminadas neste conjunto,

este valor foi escolhido pois apresentou uma alta performance para os conjuntos de teste com até 40% de contaminação.

Foram inseridos ruído impulsivo, 40% dos pixels contaminados, em 30%, 60% e 90% das imagens do conjunto de treino, as Tabelas 6, 7 e 8 relatam os resultados obtidos.

Tabela 6: Acurácias da CNN em Top-1 para o conjunto de teste de imagens com ruído de impulso (sal e pimenta) e treinada com 30% das imagens do conjunto de treino contaminadas com o mesmo ruído, sendo 40% dos pixels dessas imagens contaminados.

% pixels contaminados	0	5	10	15	20	40	60	80	100
Acurácia	98,44	98,09	97,89	97,35	96,70	91,84	71,22	31,36	9,43

Fonte: Elaborada pelo autor.

Tabela 7: Acurácias da CNN em Top-1 para o conjunto de teste de imagens com ruído de impulso (sal e pimenta) e treinada com 60% das imagens do conjunto de treino contaminadas com o mesmo ruído, sendo 40% dos pixels dessas imagens contaminados.

% pixels contaminados	0	5	10	15	20	40	60	80	100
Acurácia	98,11	98,80	97,77	97,50	97,08	93,88	71,50	27,86	9,96

Fonte: Elaborada pelo autor.

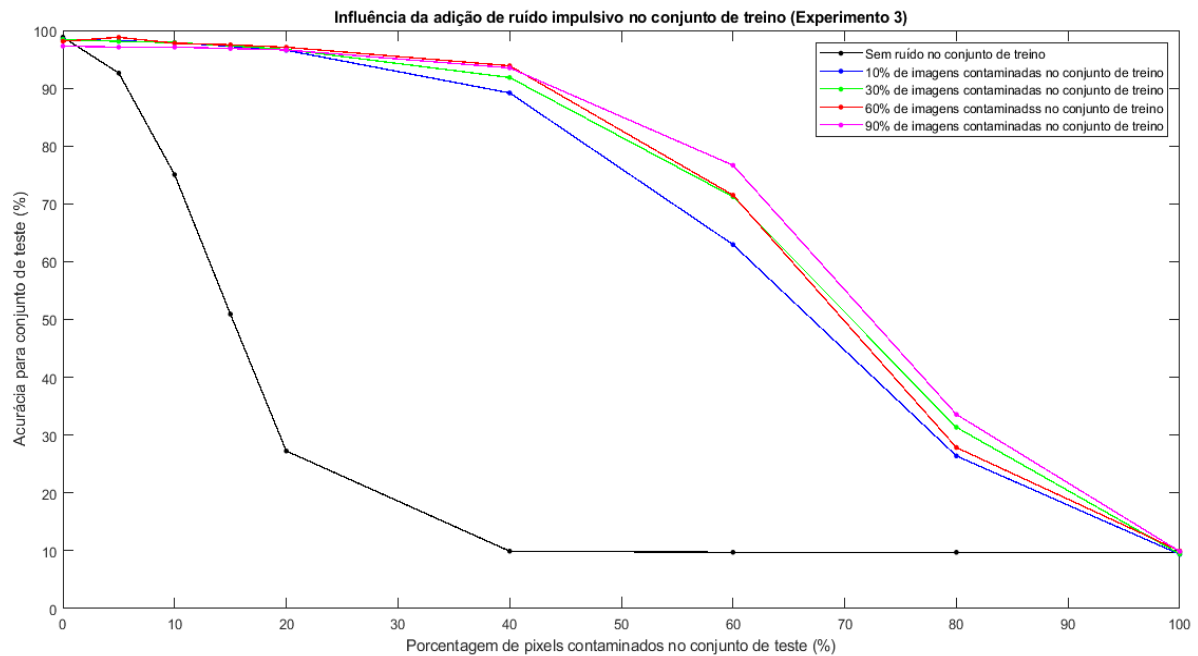
Tabela 8: Acurácias da CNN em Top-1 para o conjunto de teste de imagens com ruído de impulso (sal e pimenta) e treinada com 90% das imagens do conjunto de treino contaminadas com o mesmo ruído, sendo 40% dos pixels dessas imagens contaminados.

% pixels contaminados	0	5	10	15	20	40	60	80	100
Acurácia	97,28	97,10	97,10	96,86	96,61	93,51	76,67	33,58	9,88

Fonte: Elaborada pelo autor.

Os resultados das tabelas do Experimento 3 são exibidos na Figura 14, juntamente com os resultados do Experimento 1 (Tabela 1), e o resultado do Experimento 2 para o caso de adição de 40% de pixels contaminados no conjunto de treino (Tabela 4).

Figura 14: Desempenho da rede classificando imagens com ruído de impulso (CNN treinada com diferentes quantidades de imagens contaminadas).



Fonte: Elaborada pelo autor.

Pode-se observar que não existe grande variação na acurácia da rede, sendo a maior delas uma diferença de 13,72% entre a quantidade de 10% e 90% de imagens contaminadas no conjunto de treino para o caso de 60% de imagens contaminadas no conjunto de teste.

Observa-se que mesmo contaminando 90% das imagens de treino, a rede perdeu apenas 1,55% de acurácia em relação a rede treinada sem imagens contaminadas para o caso sem inserção de ruído no conjunto de teste. Isso demonstra que a redução a apenas 10% de imagens sem contaminação não interfere abruptamente na acurácia da rede para baixos níveis de contaminação.

Por fim, pode-se observar que os resultados com 30% de imagens contaminadas com 40% de pixels contaminados (Tabela 6) foi o caso que atingiu a melhor performance para índices de contaminação mais altos, sem perder consideravelmente sua acurácia para valores baixos de contaminação.

4.1.4 Experimento 4

A última etapa da análise do ruído impulsivo consiste na comparação com um método convencional de remoção desse tipo de contaminação: filtragem da mediana.

Para este experimento todos os conjuntos de teste contaminados com ruído foram filtrados por um filtro da mediana 3x3. Esse conjunto filtrado foi usado como teste para a rede treinada sem imagens contaminadas, os resultados são exibidos na Tabela 9.

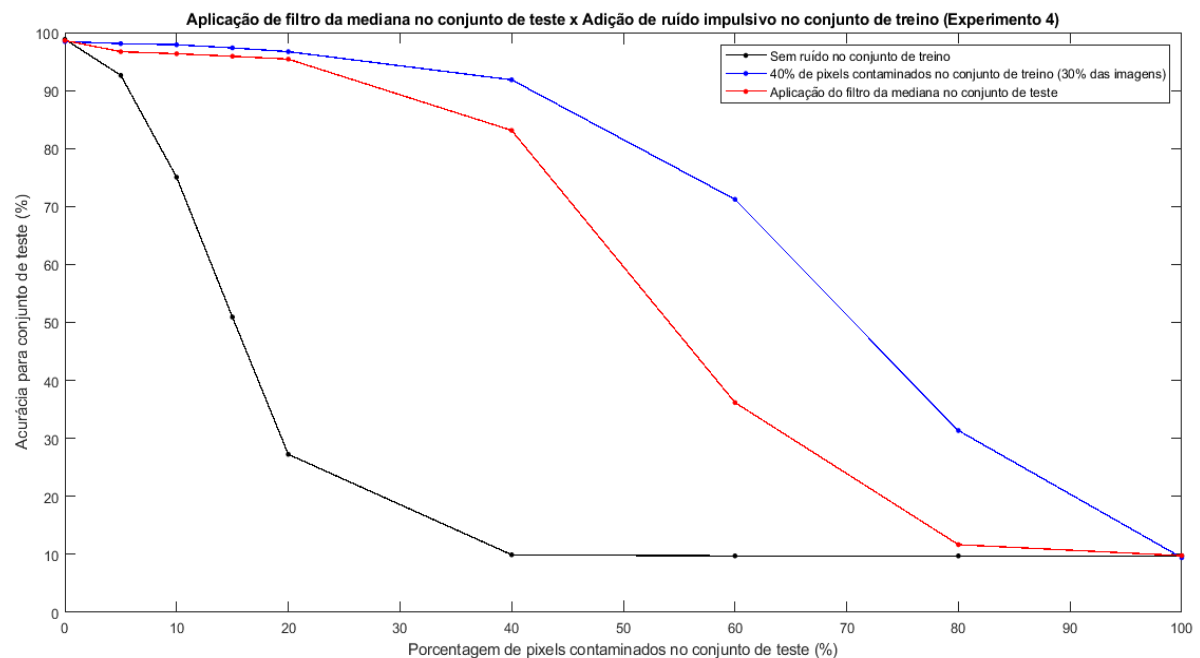
Tabela 9: Acurácias em Top-1 da CNN treinada com imagens sem ruído, para conjunto de teste de imagens com ruído de impulso (sal e pimenta) tratadas com filtro da mediana 3x3.

% pixels contaminados	0	5	10	15	20	40	60	80	100
Acurácia	98,62	96,69	96,34	95,90	95,41	83,12	36,18	11,68	9,78

Fonte: Elaborada pelo autor.

A Figura 15 exibe o resultado obtido no Experimento 1 (Tabela 1), a rede treinada que produziu a maior acurácia nos Experimentos 2 e 3 (30% de imagens contaminadas com 40% pixels contaminados no conjunto de treino, Tabela 6) e os resultados exibidos na Tabela 9.

Figura 15: Comparação do desempenho da rede classificando imagens com ruído de impulso (Utilizando imagens no treino com e sem ruído, e com aplicação de filtros).



Fonte: Elaborada pelo autor.

Pode-se observar que o filtro da mediana conseguiu promover uma boa acurácia para uma contaminação de até 40% dos pixels (73,19% a mais que o

resultado do Experimento 1) exibindo uma acurácia de 83,12%, além de mantê-la alta para até 20% de pixels contaminados (95,41%).

Porém, a inserção de ruído no conjunto de treino se mostrou um método mais robusto (um classificador eficiente para uma contaminação de até 40% nos pixels), com uma diferença de até 35,04% para o caso de 60% de pixels contaminados no conjunto de teste.

O olho humano não consegue reconhecer os dígitos filtrados somente após a inserção de 80% de ruído, 2 últimas imagens da linha inferior da Figura 7. Portanto a filtragem da mediana se mostrou o método menos eficaz, sendo que a inserção de ruído no conjunto de treino se aproximou bastante do reconhecimento do ser humano.

4.2 RUÍDO GAUSSIANO

4.2.1 Experimento 1

A Tabela 10 apresenta as acurácias obtidas para a rede treinada com o conjunto de treino MNIST e testada com o conjunto de teste MNIST com diferentes proporções de ruído gaussiano inseridos.

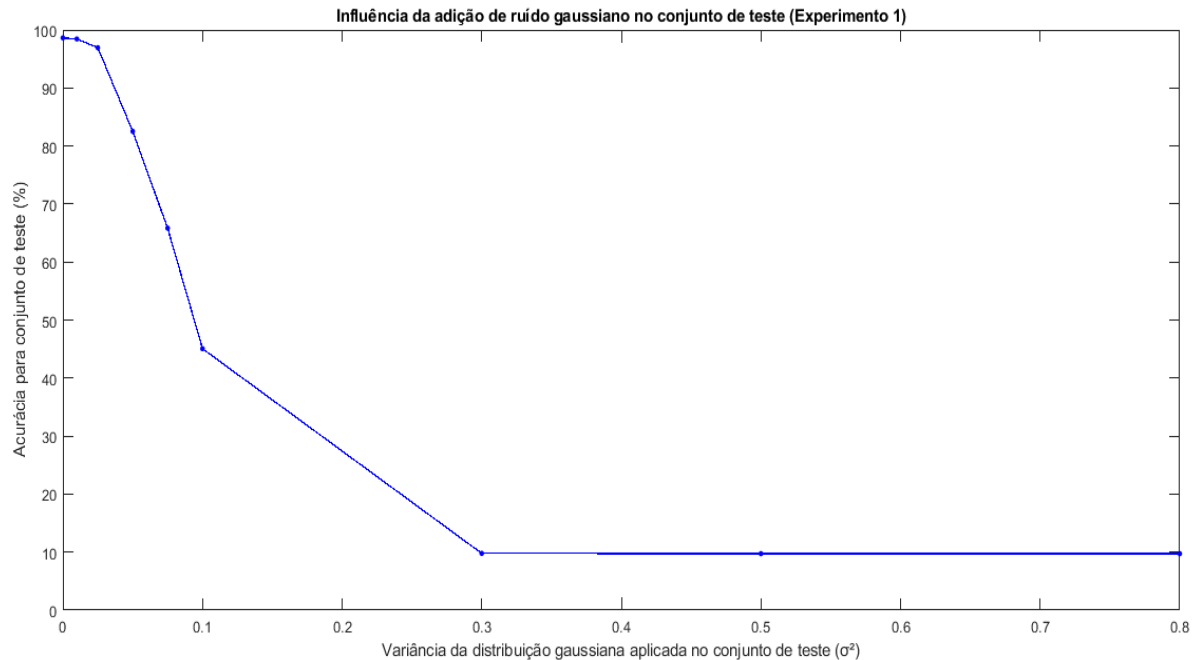
Tabela 10: Acurácias em Top-1 da CNN treinada com imagens sem ruído, para conjunto de teste de imagens com ruído Gaussiano.

Variância	0	0,01	0,025	0,05	0,075	0,1	0,3	0,5	0,8
Acurácia	98,64	98,45	96,94	82,53	65,85	45,06	9,79	9,74	9,74

Fonte: Elaborada pelo autor.

Pode-se observar que com o aumento do ruído (aumento da variância) a rede perdeu seu poder de classificação quando o ruído contaminante foi inserido com variância entre 0,075 e 0,1, obtendo uma acurácia de 65,85% e 45,06% respectivamente. Seres humanos, no entanto, não conseguem distinguir o dígito manuscrito “zero” com clareza apenas nas duas últimas imagem da linha inferior da Figura 8 (variância igual a 0,5 e 0,8). A Figura 16 demonstra os valores mostrados na Tabela 10.

Figura 16: Desempenho da rede classificando imagens com ruído gaussiano (CNN treinada com imagens sem ruído).



Fonte: Elaborada pelo autor.

4.2.2 Experimento 2

Para o Experimento 2, foram utilizados os valores de variância no conjunto de teste a partir do ponto em que a rede perdeu seu poder de classificação. Assim foram escolhidos os valores de variância igual a 0,075; 0,1; 0,3 e 0,8 para contaminar 10% das imagens do conjunto de treino, nas Tabelas 11, 12, 13 e 14 estão os resultados obtidos para cada caso, respectivamente.

Tabela 11: Acurácias da CNN em Top-1 para conjunto de teste de imagens com ruído Gaussiano e treinada com 10% das imagens contaminadas com o mesmo ruído, utilizando variância igual a 0,075.

Variância	0	0,01	0,025	0,05	0,075	0,1	0,3	0,5	0,8
Acurácia	98,87	98,75	98,50	98,08	97,74	96,92	79,43	58,59	40,18

Fonte: Elaborada pelo autor.

Tabela 12: Acurácias da CNN em Top-1 para conjunto de teste de imagens com ruído Gaussiano e treinada com 10% das imagens contaminadas com o mesmo ruído, utilizando variância igual a 0,1.

Variância	0	0,01	0,025	0,05	0,075	0,1	0,3	0,5	0,8
Acurácia	98,73	98,62	98,46	98,32	97,76	97,17	75,04	52,24	33,55

Fonte: Elaborada pelo autor.

Tabela 13: Acurácias da CNN em Top-1 para conjunto de teste de imagens com ruído Gaussiano e treinada com 10% das imagens contaminadas com o mesmo ruído, utilizando variância igual a 0,3.

Variância	0	0,01	0,025	0,05	0,075	0,1	0,3	0,5	0,8
Acurácia	98,45	98,28	98,13	97,8	97,18	97,11	89,70	81,11	67,96

Fonte: Elaborada pelo autor

Tabela 14: Acurácias da CNN em Top-1 para conjunto de teste de imagens com ruído Gaussiano e treinada com 10% das imagens contaminadas com o mesmo ruído, utilizando variância igual a 0,8.

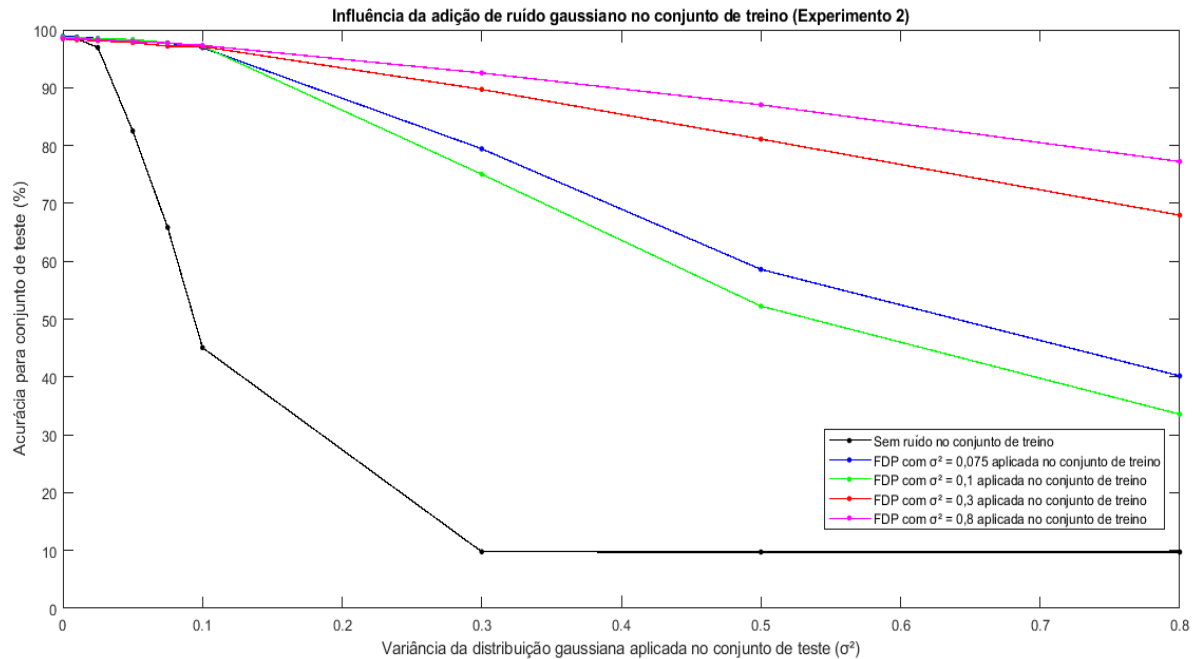
Variância	0	0,01	0,025	0,05	0,075	0,1	0,3	0,5	0,8
Acurácia	98,52	98,46	98,29	98,00	97,71	97,3	92,54	87,04	77,23

Fonte: Elaborada pelo autor.

Pode-se observar que, da mesma maneira que nos experimentos com ruído impulsivo, a rede aprendeu com o ruído Gaussiano inserido no treinamento e aumentou seu desempenho em até 82% para os melhores casos, como pode ser visto na Figura 17. A acurácia obtida com imagens de teste contaminadas com ruído Gaussiano com variância de 0,3 subiu de 9,79% (Tabela 10) para 92,54% (Tabela 14). Isso corresponde dizer que a rede treinada com 10% de imagens contaminadas com ruído Gaussiano de variância 0,8 foi capaz de identificar com acurácia média superior a 92,54% todas as imagens da linha superior da Figura 8 e mais a primeira e segunda imagem da linha inferior. É importante salientar que a segunda imagem da linha inferior da Figura 8 pertence a um universo que foi discriminado pela CNN com cerca de 75% de acurácia para o pior. Isso se aproxima do reconhecimento realizado por seres humanos quando analisam as imagens ruidosas da Figura 8.

Por fim, pode-se observar que as redes treinadas com variância igual a 0,3 e 0,8 demonstraram uma acurácia maior comparadas às demais redes do Experimento 2. Exibindo uma diferença de até 34,8% para o caso de 0,5 de variância no conjunto de teste (comparação entre rede treinada com 0,8 de variância e 0,1 de variância).

Figura 17: Desempenho da rede classificando imagens com ruído gaussiano (CNN treinada com imagens com ruído em diferentes proporções).



Fonte: Elaborada pelo autor.

4.2.3 Experimento 3

Para o terceiro experimento foi fixada a variância igual a 0,3 para o conjunto de treino, e alterado a quantidade de imagens contaminadas, as Tabelas 15, 16 e 17 mostram os resultados para 30%, 60% e 90% de imagens contaminadas com 0,3 de variância no conjunto de treino.

Tabela 15: Acurácias da CNN em Top-1 para conjunto de teste de imagens com ruído Gaussiano e treinada com 30% das imagens contaminadas com o mesmo ruído, utilizando variância igual a 0,3.

Variância	0	0,01	0,025	0,05	0,075	0,1	0,3	0,5	0,8
Acurácia	98,50	98,43	98,22	98,05	97,94	97,94	93,59	86,44	73,14

Fonte: Elaborada pelo autor.

Tabela 16: Acurácias da CNN em Top-1 para conjunto de teste de imagens com ruído Gaussiano e treinada com 60% das imagens contaminadas com o mesmo ruído, utilizando variância igual a 0,3.

Variância	0	0,01	0,025	0,05	0,075	0,1	0,3	0,5	0,8
Acurácia	97,98	97,91	97,87	97,81	97,57	97,64	93,95	88,49	77,06

Fonte: Elaborada pelo autor.

Tabela 17: Acurácias da CNN em Top-1 para conjunto de teste de imagens com ruído Gaussiano e treinada com 90% das imagens contaminadas com o mesmo ruído, utilizando variância igual a 0,3.

Variância	0	0,01	0,025	0,05	0,075	0,1	0,3	0,5	0,8
Acurácia	97,78	97,73	97,72	97,55	97,57	97,39	94,66	87,86	70,17

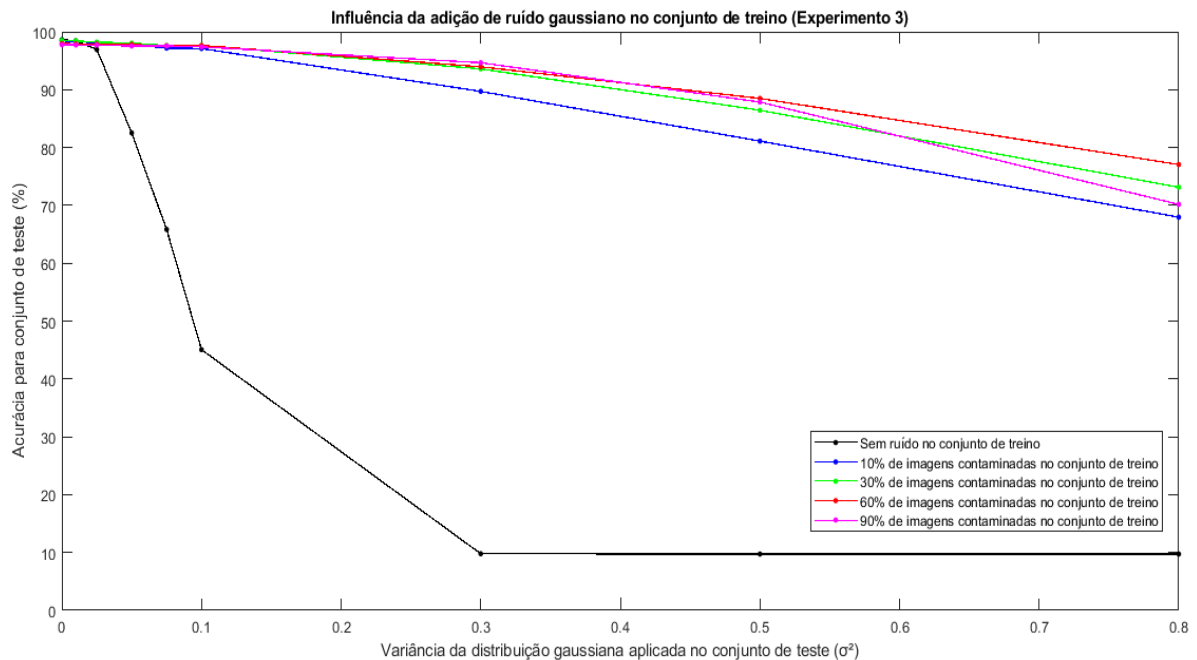
Fonte: Elaborada pelo autor.

Assim como nos resultados obtidos para o ruído impulsivo, não houve grande variação na acurácia para alterações na quantidade de imagens contaminadas no conjunto de treino, a Figura 18 demonstra os resultados obtidos nesse experimento. Atingindo uma diferença máxima de 7,38% entre a acurácia correspondente a 60% e 10% das imagens contaminadas no conjunto de treino para o caso de 0,5 de variância no conjunto de teste.

A acurácia da rede sem contaminação no treino (Experimento 1, Tabela 10) para o conjunto de teste sem contaminação caiu apenas 0,86% com relação ao conjunto de treino com 90% das imagens contaminadas, evidenciando que, também para a adição do ruído gaussiano, a redução drástica de imagens sem contaminação no conjunto de treino não afeta consideravelmente a acurácia da rede para imagens sem contaminação.

Por fim, pode-se dizer que a inserção de imagens contaminadas com ruído gaussiano de variância igual a 0,3 em 60% das imagens no conjunto de treino (Tabela 16) gerou a rede que obteve melhor eficácia, obtendo uma diferença máxima de 84,16% com relação ao Experimento 1 (Tabela 10) para o caso do conjunto de teste com variância igual a 0,3

Figura 18: Desempenho da rede classificando imagens com ruído gaussiano (CNN treinada com diferentes quantidades de imagens contaminadas).



Fonte: Elaborada pelo autor.

4.2.4 Experimento 4

A última etapa da análise do ruído gaussiano consiste na comparação com um método convencional de remoção desse tipo de ruído: filtragem da média.

Para este experimento todos os conjuntos de teste contaminados com ruído foram filtrados por um filtro da média normalizado 3x3. Esses conjuntos filtrados foram usados como teste para a rede treinada sem imagens contaminadas no conjunto de treino, os resultados são exibidos na Tabela 18.

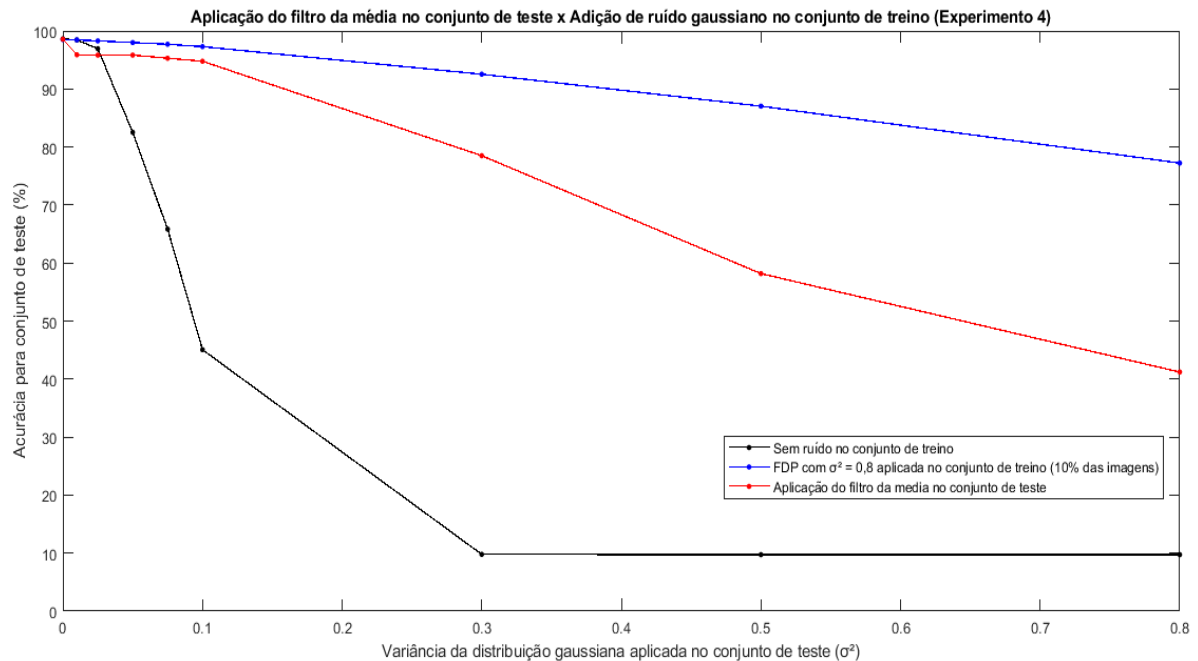
Tabela 18: Acurácias em Top-1 da CNN treinada com imagens sem ruído, para conjunto de teste de imagens com ruído de gaussiano tratadas com filtro da média 3x3.

Variância	0	0,01	0,025	0,05	0,075	0,1	0,3	0,5	0,8
Acurácia	98,58	95,89	95,83	95,84	95,30	94,78	78,50	58,18	41,21

Fonte: Elaborada pelo autor.

A Figura 19 exibe os resultados do Experimento 4, juntamente com os resultados do Experimento 1 (Tabela 10) e também a solução que apresentou os melhores valores de acurácia para contaminação gaussiana (10% das imagens contaminadas no conjunto de treino com variância igual a 0,8, Tabela 14).

Figura 19: Comparação do desempenho da rede classificando imagens com ruído gaussiano (Utilizando imagens no treino com e sem ruído, e com aplicação de filtros).



Fonte: Elaborada pelo autor.

O filtro da média se mostrou uma boa performance para uma contaminação de até 0,1 de variância, 94,78% de acurácia (49,72% a mais que os resultados de Experimento 1).

Para os primeiros valores de variância, até 0,1, o filtro da média se mostrou quase tão eficaz quanto a adição de imagens ruidosas no conjunto de treino, permanecendo apenas uma diferença de cerca de 2% de acurácia. No entanto para os valores mais altos, acima de 0,1, a solução por meio da filtragem se mostrou consideravelmente menos efetiva, até 36,02% menor para o caso de 0,8 de variância no conjunto de teste quando comparado à contaminação de 10% das imagens no conjunto de treino com 0,8 de variância (classificando com eficácia um ruído com variância igual a 0,3, Tabela 14).

O olho humano não reconhece facilmente somente as duas últimas imagens filtradas referente à Figura 9. Portanto, pode-se afirmar que a filtragem da média não foi uma solução que se aproximasse ao potencial do olho humano. No entanto, a inserção de ruído no conjunto de treino se mostrou eficaz ao ponto de poder ser comparada aos seres humanos.

4.3 RUÍDO SPECKLE

4.3.1 Experimento 1

Na última seção de experimentos, foram realizados testes com ruído do tipo *speckle*. Abaixo está a Tabela 19 com as acurácias obtidas para a rede treinada com o conjunto de treino MNIST e testada com o conjunto de teste MNIST com diferentes proporções de ruído do tipo *speckle* inseridos.

Tabela 19: Acurácias em Top-1 da CNN treinada com imagens sem ruído, para conjunto de teste de imagens com ruído do tipo *speckle*.

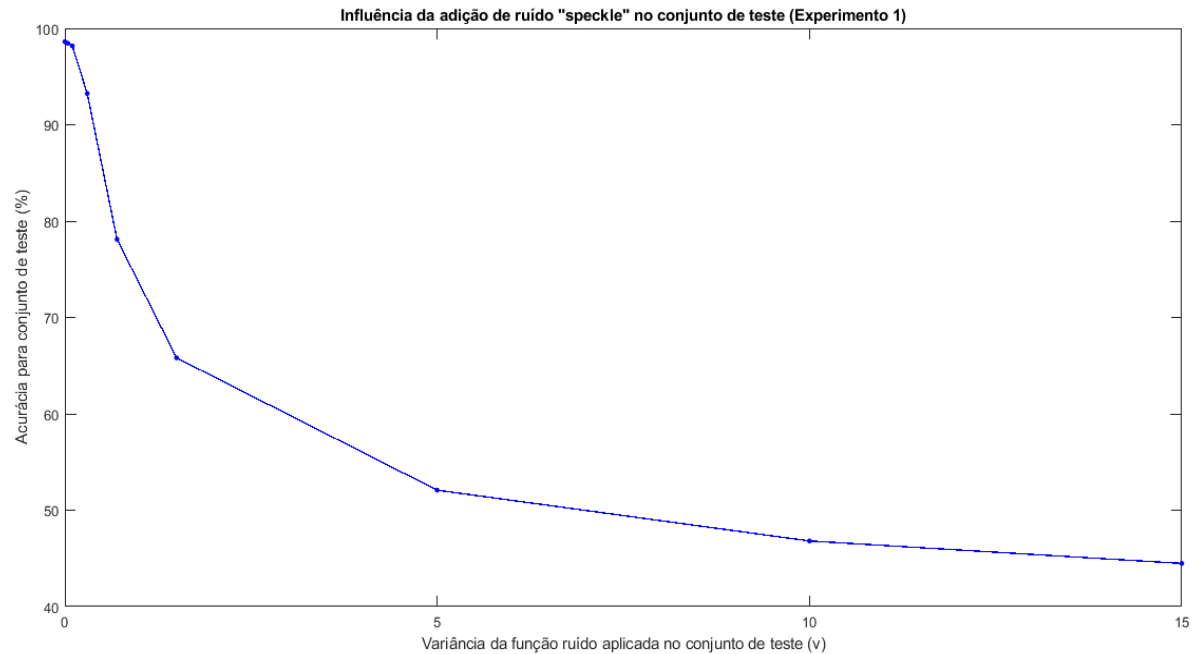
Variância	0	0,04	0,1	0,3	0,7	1,5	5	10	15
Acurácia	98,62	98,45	98,18	93,24	78,11	65,81	52,08	46,81	44,48

Fonte: Elaborada pelo autor.

A Figura 20 exibe os resultados da Tabela 19. Pode-se observar que a rede reconhece com uma acurácia aceitável até a inserção de ruído com variância igual a 0,3 (93,24%). Porém, a partir desse ponto ela decresce e alcança um mínimo de 44,48%.

Devido às características deste tipo de ruído, tipo multiplicativo, as imagens dos dígitos não se deterioram ao ponto de ficarem irreconhecíveis ao olho humano, pois o fundo preto (intensidade 0) não é afetado. Ocorre apenas a interferência nos pixels mais claros, sendo que nos casos mais extremos grande parte dos pixels saturam (ficam totalmente brancos ou pretos). Assim, o olho humano é capaz de reconhecer todos os dígitos da Figura 10.

Figura 20: Desempenho da rede classificando imagens com ruído *speckle* (CNN treinada com imagens sem ruído).



Fonte: Elaborada pelo autor.

4.3.2 Experimento 2

Para o Experimento 2, foram utilizados os valores de variância no conjunto de teste a partir do ponto em que a rede perdeu seu alto poder de classificação. Assim foram escolhidos os valores de variância igual a 0,7; 1,5; 5 e 15 para contaminar 10% da imagens do conjunto de teste, nas Tabelas 20, 21, 22 e 23 abaixo estão os resultados obtidos para cada caso, respectivamente.

Tabela 20: Acurácias da CNN em Top-1 para conjunto de teste de imagens com ruído *speckle* e treinada com 10% das imagens contaminadas com o mesmo ruído, utilizando variância igual a 0,7.

Variância	0	0,04	0,1	0,3	0,7	1,5	5	10	15
Acurácia	98,54	98,37	98,34	97,69	96,02	94,12	92,20	90,53	90,14

Fonte: Elaborada pelo autor.

Tabela 21: Acurácias da CNN em Top-1 para conjunto de teste de imagens com ruído *speckle* e treinada com 10% das imagens contaminadas com o mesmo ruído, utilizando variância igual a 1,5.

Variância	0	0,04	0,1	0,3	0,7	1,5	5	10	15
Acurácia	98,73	98,50	98,24	97,78	96,85	95,99	94,43	93,61	93,04

Fonte: Elaborada pelo autor.

Tabela 22: Acurácias da CNN em Top-1 para conjunto de teste de imagens com ruído *speckle* e treinada com 10% das imagens contaminadas com o mesmo ruído, utilizando variância igual a 5.

Variância	0	0,04	0,1	0,3	0,7	1,5	5	10	15
Acurácia	98,74	98,64	98,47	98,08	97,36	96,76	95,79	95,66	95,12

Fonte: Elaborada pelo autor

Tabela 23: Acurácias da CNN em Top-1 para conjunto de teste de imagens com ruído *speckle* e treinada com 10% das imagens contaminadas com o mesmo ruído, utilizando variância igual a 15.

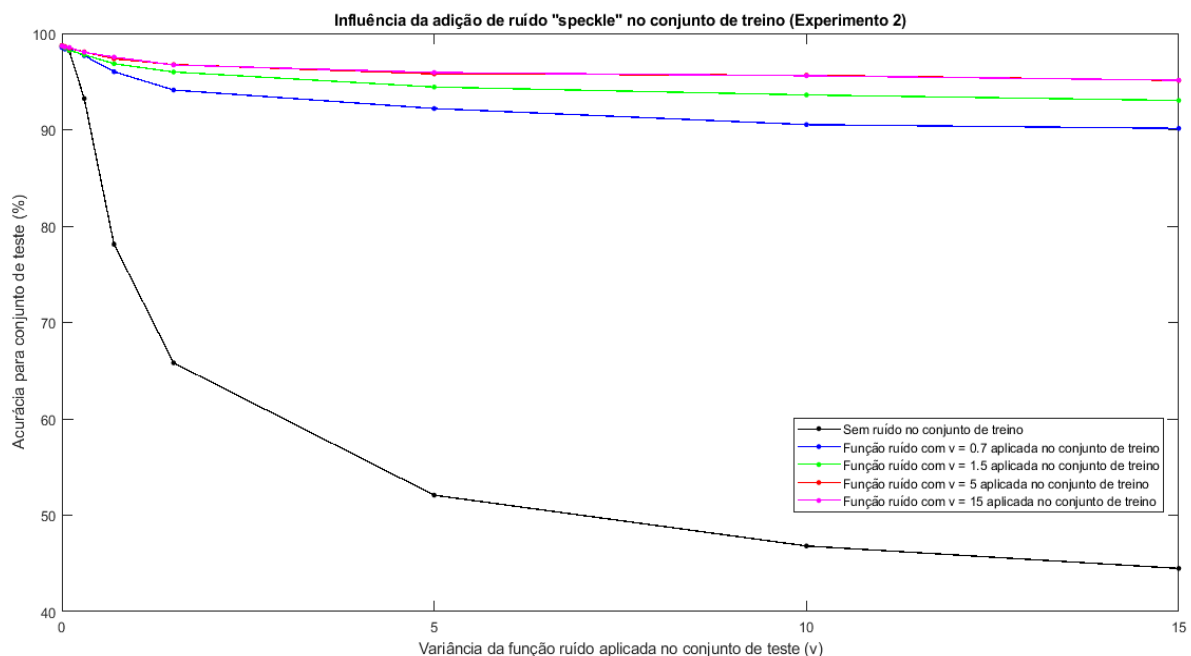
Variância	0	0,04	0,1	0,3	0,7	1,5	5	10	15
Acurácia	98,70	98,59	98,54	98,04	97,54	96,72	95,94	95,60	95,14

Fonte: Elaborada pelo autor

As tabelas acima estão demonstradas na Figura 21, juntamente com o resultado obtido no Experimento 1 (Tabela 19). Todos os casos apresentaram uma melhora significativa na acurácia da CNN, sendo que os experimentos referentes às Tabelas 22 e 23 (variância igual a 5 e a 15) obtiveram a melhor performance, conseguindo um acréscimo de até 50,66% de acurácia, conjunto de teste contaminado com variância igual a 15, quando comparado ao Experimento 1 (Tabela 19).

Portanto, para este caso, este método para solucionar a interferência de ruído do tipo *speckle* demonstrou uma boa performance para todos os valores da variância.

Figura 21: Desempenho da rede classificando imagens com ruído *speckle* (CNN treinada com imagens com ruído em diferentes proporções).



Fonte: Elaborada pelo autor.

4.3.3 Experimento 3

No Experimento 3 foi observado a interferência que a alteração da quantidade de imagens contaminadas no conjunto de treino causa. Para este exercício foi fixada a variância no valor de 0,7 e alterada a quantidade de imagens contaminadas no conjunto de treino. As Tabelas 24, 25 e 26 indicam as acurácias obtidas para as proporções de imagens contaminadas iguais a 30%, 60% e 90%, respectivamente.

Tabela 24: Acurácias da CNN em Top-1 para conjunto de teste de imagens com ruído *speckle* e treinada com 30% das imagens contaminadas com o mesmo ruído, utilizando variância igual a 0,7.

Variância	0	0,04	0,1	0,3	0,7	1,5	5	10	15
Acurácia	98,68	98,58	98,45	98,09	97,36	96,38	95,64	94,75	94,65

Fonte: Elaborada pelo autor.

Tabela 25: Acurácias da CNN em Top-1 para conjunto de teste de imagens com ruído *speckle* e treinada com 60% das imagens contaminadas com o mesmo ruído, utilizando variância igual a 0,7.

Variância	0	0,04	0,1	0,3	0,7	1,5	5	10	15
Acurácia	98,34	98,38	98,35	98,07	97,70	97,23	96,19	95,67	95,38

Fonte: Elaborada pelo autor.

Tabela 26: Acurácias da CNN em Top-1 para conjunto de teste de imagens com ruído *speckle* e treinada com 90% das imagens contaminadas com o mesmo ruído, utilizando variância igual a 0,7.

Variância	0	0,04	0,1	0,3	0,7	1,5	5	10	15
Acurácia	98,73	98,58	98,65	98,49	98,01	97,36	96,61	95,96	95,34

Fonte: Elaborada pelo autor.

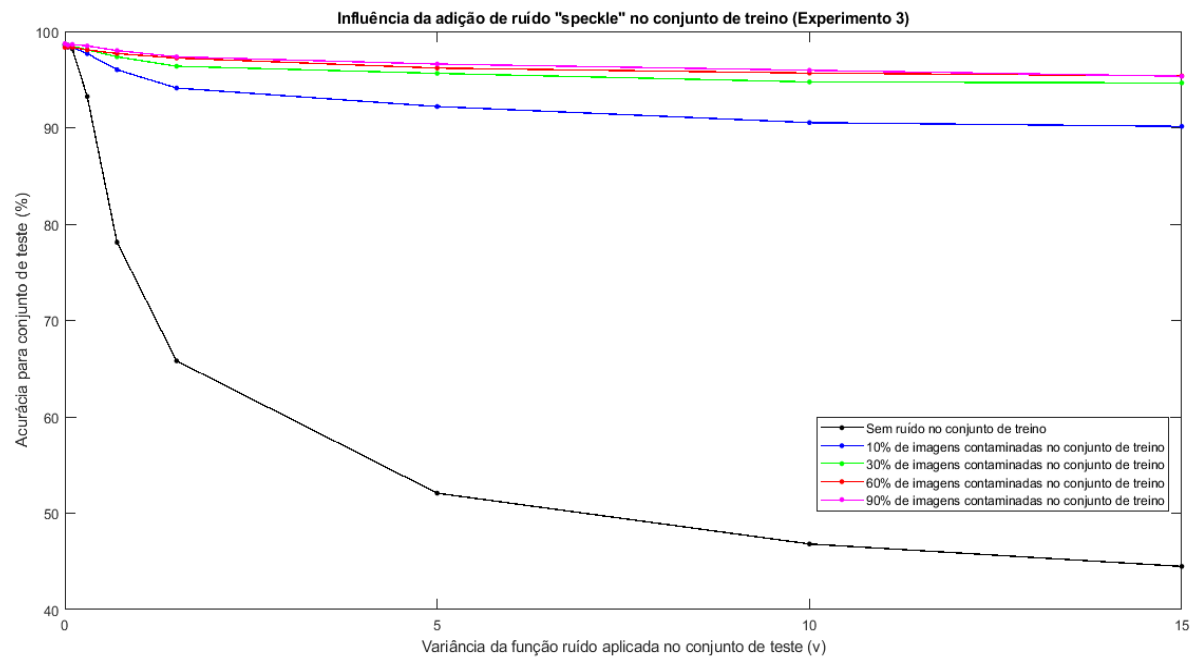
Na Figura 22 estão demonstrados os valores obtidos no Experimento 3, juntamente com os resultados de 10% das imagens contaminadas com variância igual a 0,7, referente a Tabela 20 do Experimento 2, e também com os resultados do Experimento 1, Tabela 19.

Nota-se que com o aumento da quantidade de imagens contaminadas no conjunto de treino, a acurácia foi levemente aperfeiçoada, aumentando aproximadamente em até 5% os valores da Tabela 20 para os casos com variância igual a 10 e 15, porém essa variação não afetou significativamente as acurácias encontradas no experimento anterior.

Condizente com o Experimento 2, a rede continuou exibindo uma alta acurácia até mesmo para o pior caso, variância igual a 15, obtendo os melhores valores para

90% das imagens contaminadas no treino. Portanto pode-se dizer que a CNN treinada com ruído pode reconhecer todas as imagens da Figura 10, assim como o olho humano.

Figura 22: Desempenho da rede classificando imagens com ruído *speckle* (CNN treinada com diferentes quantidades de imagens contaminadas).



Fonte: Elaborada pelo autor.

4.3.4 Experimento 4

A última etapa da análise do ruído *speckle* consiste na comparação com um método convencional de remoção desse tipo de ruído: filtragem da média.

Para este experimento, todos os conjuntos de teste contaminados com ruído foram filtrados por um filtro da média normalizado 3x3. Esse conjunto filtrado foi usado como teste para a rede treinada sem imagens contaminadas no conjunto de treino, os resultados são exibidos na Tabela 27.

Tabela 27: Acurácias em Top-1 da CNN treinada com imagens sem ruído, para conjunto de teste de imagens com ruído de *speckle* tratadas com filtro da média 3x3.

Variância	0	0,04	0,1	0,3	0,7	1,5	5	10	15
Acurácia	98,65	94,80	93,58	88,83	79,86	72,54	65,93	63,34	62,47

Fonte: Elaborada pelo autor.

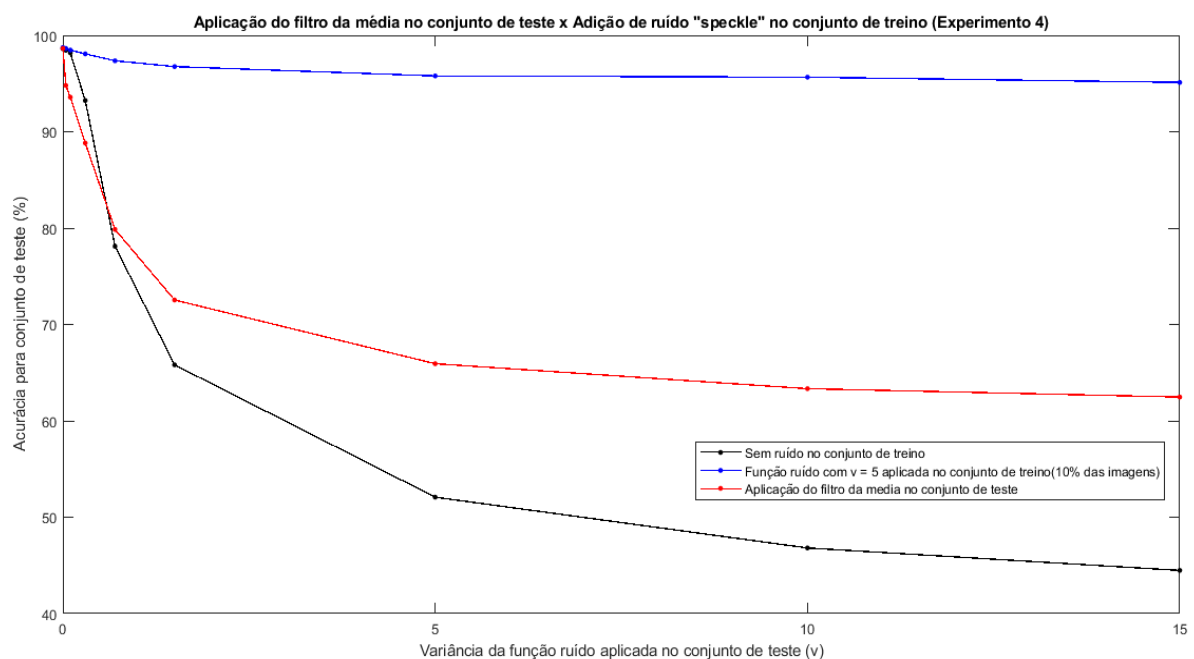
A Figura 23 exibe os resultados do Experimento 4, juntamente com os resultados do Experimento 1 (Tabela 19) e também os resultados que exibiram a melhor performance para contaminação com ruído do tipo *speckle* (10% das imagens contaminadas no conjunto de treino com variância igual a 5, Tabela 22).

A filtragem da média no conjunto de teste melhora a performance da rede para os casos de grande quantidade de ruído inseridas no conjunto de treino, até 17,99% (variância igual a 15 no conjunto de teste) porém piora para os casos de pouco ruído inserido, ruído com variância menor que 0,3 no conjunto de teste (cerca de 5% pior).

Além disso, este método se mostra menos eficaz que a inserção de imagens contaminadas no conjunto de treino, possuindo uma diferença de até 32,65%, variância igual a 15 no conjunto de teste, para o caso de 10% de imagens contaminadas com ruído com variância igual a 5 no treino (Tabela 22).

Seres humanos são capazes de reconhecer todas as imagens da Figura 10 e 11, sendo equiparado a inserção de ruído no conjunto de treino, sendo um bom classificador para todas as proporções de ruído. Porém, a filtragem da média não se mostrou capaz de alcançar o olho humano.

Figura 23: Comparação do desempenho da rede classificando imagens com ruído *speckle* (Utilizando imagens no treino com e sem ruído, e com aplicação de filtros).



Fonte: Elaborada pelo autor.

5 CONCLUSÃO

Este trabalho fez uma análise do desempenho de Redes Neurais Convolucionais para o reconhecimento de dígitos manuscritos quando as imagens são degradadas com ruído. Três tipos de ruídos, dois aditivos e um multiplicativo foram inseridos isoladamente, o primeiro um ruído de impulso (sal e pimenta) que é um típico ruído contaminante devido a problemas na fonte de aquisição de imagem, componentes defeituosos ou superaquecidos. O segundo ruído utilizado foi o Gaussiano, que é um ruído estatístico que contamina imagens digitais durante a aquisição no sensor causado por iluminação fraca e/ou alta temperatura e também durante a transmissão, gerado pelo circuito eletrônico. O terceiro ruído utilizado foi o *speckle*, que surge durante a aquisição de imagens, devido ao efeito das condições ambientais nos sensores.

Os experimentos realizados mostraram que se a CNN for treinada apenas com imagens sem ruído seu desempenho de classificação cai drasticamente tanto para conjunto de teste com imagens degradadas com ruído de impulso como para conjunto de teste com imagens degradadas com ruído Gaussiano ou *speckle*. Este comportamento é bastante inferior à capacidade humana de discriminação de imagens degradadas por ruído. No entanto, se forem inseridas imagens contaminadas no conjunto de treinamento a CNN aprende e tem um ganho superior a 83% de acurácia no reconhecimento de imagens degradadas por ruído de impulso, superior a 84% de acurácia no reconhecimento de imagens degradadas por ruído Gaussiano e superior a 50% de acurácia no reconhecimento de imagens degradadas por ruído *speckle*. Quando as imagens de teste foram tratadas previamente utilizando-se métodos de filtragem obteve-se um aumento da acurácia de até 73% para imagens degradadas com ruído de impulso, 49% para imagens degradadas com ruído Gaussiano e 17% para imagens degradadas com ruído *speckle*. Apesar do aumento de acurácia que a filtragem oferece, este método não se aproxima do reconhecimento do olho humano. No entanto o método de inserção de ruído no conjunto de treino se mostrou mais eficiente que o método de filtragem e aproximou-se bastante da capacidade humana de discriminação de imagens corrompidas por ruído.

REFERÊNCIAS BIBLIOGRÁFICAS

DODGE, Samuel; KARAM, Lina. Understanding how image quality affects deep neural networks. In: QUALITY OF MULTIMEDIA EXPERIENCE (QOMEX), **Eighth International Conference on IEEE**, 2016. p. 1-6.

HE, Kaiming et al. Deep residual learning for image recognition. In: THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION. **Proceedings...** 2016. p. 770-778.

HUBEL, David H.; WIESEL, Torsten N. Receptive fields and functional architecture of monkey striate cortex. **The Journal of physiology**, v. 195, n. 1, p. 215-243, 1968.

IOFFE, Sergey; SZEGEDY, Christian. Batch normalization: Accelerating deep network training by reducing internal covariate shift. **arXiv preprint arXiv:1502.03167**, 2015.

KARAHAN, Samil et al. How image degradations affect deep cnn-based face recognition?. In: BIOMETRICS SPECIAL INTEREST GROUP (BIOSIG). **International Conference of the IEEE**, 2016. p. 1-5.

KAUR, Sarbjit; SINGH, Er Ram. Image De-Noising Techniques: A Review Paper. **Infinite Study**, 2015.

KHAW, Hui Ying et al. Image noise types recognition using convolutional neural network with principal components analysis. **IET Image Processing**, v. 11, n. 12, p. 1238-1245, 2017.

KRIZHEVSKY, Alex; SUTSKEVER, Ilya; HINTON, Geoffrey E. Imagenet classification with deep convolutional neural networks. In: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS. **Proceedings...** 2012. p. 1097-1105.

LECUN, Yann et al. Backpropagation applied to handwritten zip code recognition. **Neural computation**, v. 1, n. 4, p. 541-551, 1989.

LECUN, Yann et al. Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, v. 86, n. 11, p. 2278-2324, 1998.

LECUN, Yann; CORTES, Corinna; BURGESS, Christopher J.C. The MNIST database of handwritten digits, 1998. Disponível em: <<http://yann.lecun.com/exdb/mnist/>>. Acesso em: 10 out. 2018.

LEMLEY, Joseph; CORCORAN, Peter. Transfer Learning of Temporal Information for Driver Action Classification. In: THE 28TH MODERN ARTIFICIAL INTELLIGENCE AND COGNITIVE SCIENCE CONFERENCE (MAICS), 28. **Proceedings...** 2017.

LI, Fei-Fei; KARPATHY, Andrej; JOHNSON, Justin. **Convolutional Neural Networks for Visual Recognition**: Convolutional Neural Networks (CNNs / ConvNets). Stanford University, 2017a. Disponível em: <<http://cs231n.github.io/convolutional-networks/>>. Acesso em: 14 out. 2018

LI, Fei-Fei; KARPATHY, Andrej; JOHNSON, Justin. **Convolutional Neural Networks for Visual Recognition**: Linear Classification. Stanford University, 2017b. Disponível em: <<http://cs231n.github.io/linear-classify/>>. Acesso em: 14 out. 2018

LONG, Jonathan; SHELHAMER, Evan; DARRELL, Trevor. Fully convolutional networks for semantic segmentation. In: THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION. **Proceedings...** 2015. p. 3431-3440.

NAIR, Vinod; HINTON, Geoffrey E. Rectified linear units improve restricted boltzmann machines. In: INTERNATIONAL CONFERENCE ON MACHINE LEARNING (ICML-10), 27. **Proceedings....** 2010. p. 807-814.

PANDEY, Alka; SINGH, K. K. Analysis of Noise Models in Digital Image Processing. **International Journal of Science, Technology & Management**, v. 4, n. 01, p. 140-144, 2015.

PARKHI, Omkar M. et al. Deep face recognition. **BMVC**. 2015. p. 6.

SCHERER, Dominik; MÜLLER, Andreas; BEHNKE, Sven. Evaluation of pooling operations in convolutional architectures for object recognition. **ARTIFICIAL NEURAL NETWORKS–ICANN 2010. Proceedings...** Springer, Berlin, Heidelberg, 2010. p. 92-101.

SCHROFF, Florian; KALENICHENKO, Dmitry; PHILBIN, James. Facenet: A unified embedding for face recognition and clustering. In: THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION. **Proceedings...** 2015. p. 815-823.

SIMONYAN, Karen; ZISSERMAN, Andrew. Very deep convolutional networks for large-scale image recognition. **arXiv preprint arXiv:1409.1556**, 2014.

SRIVASTAVA, Nitish et al. Dropout: a simple way to prevent neural networks from overfitting. **The Journal of Machine Learning Research**, v. 15, n. 1, p. 1929-1958, 2014.

SZEGEDY, Christian et al. Going deeper with convolutions. In: THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION. **Proceedings...** 2015. p. 1-9.

SZEGEDY, Christian; TOSHEV, Alexander; ERHAN, Dumitru. Deep neural networks for object detection. In: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS. **Proceedings...** 2013. p. 2553-2561.

VERMA, Rohit; ALI, Jahid. A comparative study of various types of image noise and efficient noise removal techniques. **International Journal of advanced research in computer science and software engineering**, v. 3, n. 10, 2013.

XIE, Junyuan; XU, Linli; CHEN, Enhong. Image denoising and inpainting with deep neural networks. In: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS. **Proceeding...** 2012. p. 341-349.

ZHANG, Kai et al. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. **IEEE Transactions on Image Processing**, v. 26, n. 7, p. 3142-3155, 2017.

APÊNDICE A - ROTINA BASE PARA TREINAR E TESTAR CNN

%Carregar base de dados para treino e teste

```
imdsTrain =
imageDatastore('C:\Users\Fabio\Desktop\TCC\DATA\MNISTTrain','IncludeSubfolders
', true, 'LabelSource', 'foldernames')
imdsTest =
imageDatastore('C:\Users\Fabio\Desktop\TCC\DATA\MNISTTest', 'IncludeSubfolders'
, true, 'LabelSource', 'foldernames')
```

%Divisão do conjunto de treino em treino e validação

```
[imdsTraining,imdsValidation] = splitEachLabel(imdsTrain,0.75,'randomize');
```

%Definição das camadas da arquitetura

```
layers = [
    imageInputLayer([28 28 1], 'Normalization','zerocenter')
    convolution2dLayer(3,25,'Padding',0)
    batchNormalizationLayer
    reluLayer
    maxPooling2dLayer(2,'Stride',2)
    convolution2dLayer(3,32,'Padding',0)
    batchNormalizationLayer
    reluLayer
    fullyConnectedLayer(10)
    softmaxLayer
    classificationLayer];
```

%Opções de treinamento

```
options = trainingOptions('sgdm','Verbose',true,'Plots','training-progress',
'ValidationData',imdsValidation,
'ValidationFrequency',50,'MaxEpochs',20,'InitialLearnRate',0.01,'MiniBatchSize',128,'
ValidationPatience',5);
```

```
%Treinamento
```

```
trainedNet = trainNetwork(imdsTraining, layers, options);
```

```
%Teste e calculo da acurácia
```

```
YPred = classify(trainedNet, imdsTest);
```

```
YTest = imdsTest.Labels;
```

```
accuracy = sum(YPred == YTest)/numel(YTest)
```

APÊNDICE B – CURVA GERADA PELO MATLAB DURANTE O TREINO

