USO DE DEEP LEARNING PARA CLASSIFICAÇÃO DE IMAGENS DE LIXO

Amanda Isabela de Campos

Relatório referente à uma das
avaliações da disciplina
COC 891 - Deep Learning

Professor: Alexandre Evsukoff

Rio de Janeiro
Janeiro de 2021

1. Introdução:

1.1 Descrição do Problema

Reciclagem é uma técnica essencial para o desenvolvimento sustentável e os atuais processos de reciclagem dependem da correta classificação do tipo de material. Segundo Yang e Thung [1] os consumidores podem ficar confusos sobre como determinar o tipo de material antes de descartar, devido à grande variedade de materiais utilizados em embalagens. Por isso a necessidade de se criar um processo automático de classificação de imagens de lixo. A principal aplicação comercial dessa técnica é receber uma imagem de um material com fundo branco de um usuário e classifica-la como: papelão, vidro, plástico, papel, metal e lixo não reciclável.

O presente trabalho tem como objetivo aplicar algoritmos de inteligência artificial como redes neurais convolucionais em um conjunto de dados conhecido como TrashNet [2], dessa forma é possível prever ou classificar o tipo de lixo que uma foto apresenta. As redes neurais convolucionais, em inglês, *convolutional neural network* ou CNN, são adotadas em problemas de visão computacional, baseado em aprendizado profundo de redes neurais. Segundo Guo *et al.* [3] uma rede neural convolucional inclui principalmente três tipos de camadas, são elas, camada convolucional, camada de pooling e camada totalmente conectada. A Fig. 1 mostra a arquitetura da rede LeNet-5 (Lecun *et al*. [4]) para reconhecimento de documentos.

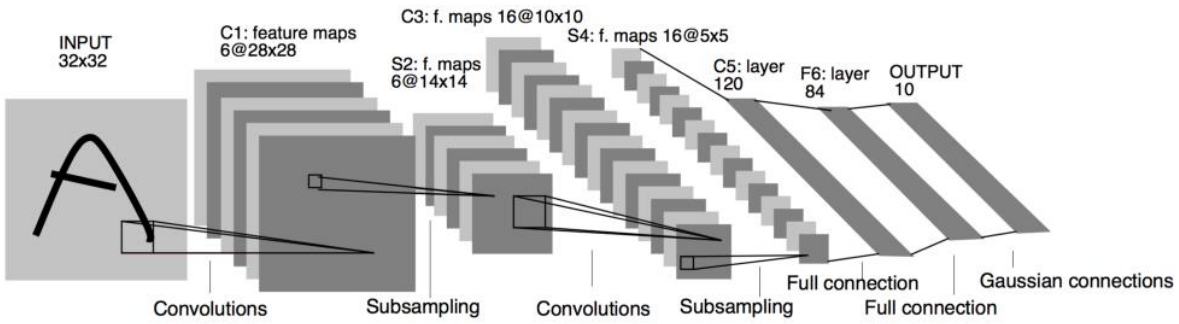

Figura 1. A arquitetura da rede LeNet-5 (Fonte: Lecun et al. [4])

O conjunto de dados adotado para o presente trabalho está disponível no repositório do github de Thung [5], composto por 2527 imagens, de tamanho 512 x 384 pixels, o que gera um dataset de 3.5 GB. As imagens são fotos, tiradas com a câmera de um celular, de lixos reais encontrados nas ruas, todas as imagens têm fundo branco e diferentes condições de iluminação. Cada imagem possui uma variável de saída que indica a qual classe pertence, sendo seis classes, distribuídas da seguinte forma: 594 imagens de papel, 501 de vidro, 137 de lixo não reciclável, 410 de metal e 482 de plástico.

1.2 Pesquisa Bibliográfica

O conjunto de dados TrashNet, adotado neste trabalho, criado por Yang e Thung [1] a partir da coleta imagens de lixo nas ruas de Stanford, todas com fundo branco, com diferentes iluminações e posições. Além disso, Yang e Thung [1] compararam modelos de SVM (*support vector machine*) e uma CNN com sete camadas, que é um modelo simplificado da rede pré-treinada AlexNet (usando 3/4 da quantidade de filtros para algumas camadas convolucionais) para classificar as imagens em seis categorias de lixo. O modelo de SVM obteve melhores resultados do que a CNN porque alcançou uma precisão de teste de 63% usando uma divisão 70/30 de treinamento/teste de dados. O SVM é um algoritmo relativamente mais simples do que a CNN, o que pode atribuir a seu sucesso nesta tarefa.

Outros pesquisadores analisaram o desempenho de classificadores com modelos redes neurais convolucionais para o mesmo conjunto de dados como Bircanoğlu *et al.* [6] que plicaram Inception-Resnet e Inception-v4 para treinamento e obteram 90% de acurácia no teste. Para transferência de aprendizagem e ajuste fino dos parâmetros de peso usando ImageNet, DenseNet121 deu o melhor resultado com 95% de precisão no teste. Além disso, propõem uma nova arquitetura especifica para classificação de imagens de materiais recicláveis, o RecycleNet e compara diferentes métodos de otimização como Adam e Adadelta.

Aral *et al.* [7] adotou modelos pré-treinados como Densenet121, DenseNet169, InceptionResnetV2, MobileNet e Xception, com os otimizadores Adam e Adadelta. Em todos os casos o otimizador Adam resultou em melhores resultados. O melhor resultado foi obtido com o DenseNet121 com acurácia de 89% e InceptionResNetV2 com acurácia de 89% com 150 e 100 épocas respectivamente mostrando a aplicabilidade de modelos de redes neurais profundas para a classificação de imagens de lixo.

Ozkaya e Seyfi [8] aplicaram as arquiteturas Alexnet, VGG16, Googlenet e Resnet para a classificação de imagens de lixo. Os resultados para 100 épocas foram Inception ResNetV2 acurácia do teste 90%, DenseNet121 e DenseNet201: acurácia do teste 85%, MobileNet: 76% com 500 épocas.

Wang *et al.* [9] utilizaram o mesmo conjunto de dados com inclusão de novas imagens e o aprendizado de transferência de uma CNN pré-treinada Resnet-50 para completar a extração de recursos. Na aplicação, é utilizado uma segmentação de imagem na fase de pré-classificação. Na etapa de pós-classificação, os pontos de amostra rotulados são integrados com o Clustering Gaussiano para localizar o objeto. O resultado modelo alcançou uma taxa de detecção total de 48,4% em precisão de simulação e classificação final de 92,4%.

2. Tecnologia:

2.1 Apresentação da tecnologia

A análise e caracterização dos dados, bem como a aplicação dos modelos de inteligência artificial serão implementadas com a linguagem de programação Python, por ser: (i) uma linguagem de programação simples, livre e aberta; (ii) a linguagem mais usada atualmente e (iii) composta de várias bibliotecas desenvolvidas e em constante atualização já implementadas para aplicações de inteligência artificial. No presente trabalho adotou-se as bibliotecas Numpy [10] (para cálculos numéricos e operações com matrizes), Pandas [11] (para manipulação de dataset em formato de tabelas e planilhas), Matplotlib [12] (para a geração de gráficos e visualização dos dados), Seaborn [13] (também para a geração de gráficos, baseado no matplotlib porém mais voltado para estatística), ScikitLearn [14] (biblioteca de inteligência artificial, com modelos já implementados de classificação) e TensorFlow [15] (biblioteca com as camadas das CNN já implementadas).

2.2 Apresentação / Visualização de Dados

A Fig. 2 apresenta o nome, o tipo e a descrição de exemplos de cada uma das 6 classes. Com o pré-processamento foi observado que o dataset não possui arquivos que não são imagens, ou seja, todos os registros serão utilizados. O conjunto de dados possui 2527 imagens, com seis classes: vidro, papel, papelão, plástico, metal e lixo.

Papelão (*Cardboard*) Vidro (*Glass*) Metal (*Metal*)

Papel (*Paper*) Plástico (*Plastic*) Lixo (*Trash*)

Figura 2. Exemplo de uma imagem de cada classe do conjunto de dados

A Fig. 3 apresenta uma visualização inicial da distribuição das classes no conjunto de dados onde observa-se o desbalanceamento entre as classes do dataset, onde a classe "*trash*" é a que possui menos dados e a classe "*paper*" possui mais imagens.

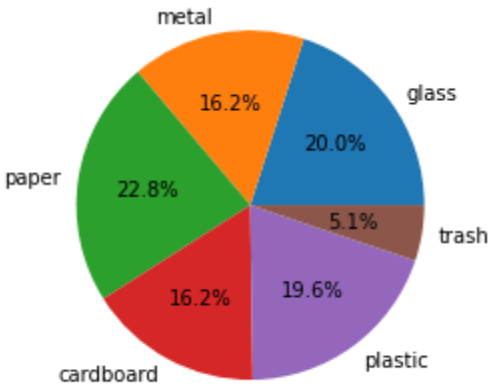

Figura 3. Divisão das classes no conjunto de dados

3. Metodologia:

A metodologia deste trabalho consiste basicamente em comparar diferentes arquiteturas de Redes Neurais convolucionais para classificação de imagens. Na etapa de pré-processamento será realizada a normalização os dados com a divisão por 255 uma vez que os valores de cada pixel são RGB variando de 0 a 255. Todas as imagens serão redimensionadas para o tamanho (150x150).

A separação entre os conjuntos de dados para treino, teste e validação foi realizada na proporção 70/15/15, respectivamente, e de forma que a mesma divisão seja usada em todas as analises, favorecendo assim que a comparação entre os modelos seja devido a arquitetura de cada modelo e não influenciada pelos dados.

Para codificar os label que indicam qual classe cada foto pertence como números inteiros é adotado o comando *LabelEncoder* e a seguir são convertidos os números inteiros em variáveis do tipo dummy. Existem diferentes maneiras de realizar essa tarefa, a mais comum e também adotada neste trabalho é a *OneHotEncoder* onde, cada variável categórica é mapeada para um vetor que contém 1 e 0 denotando a presença ou ausência do recurso. O número de vetores depende do número de categorias para as características.

A técnica de *Image Aumentation* também será aplicada para cada imagem devido ao pequeno tamanho, em questão de número de imagens, de cada classe. Essa técnica realiza a rotação, translação e zoom das imagens, criando mais imagens para o conjuto de treinamento. Segundo Yang e Thung [1]

estas transformações de imagem são importantes para levar em conta as diferentes orientações do material reciclado e para maximizar o tamanho do conjunto de dados.

Neste trabalho, portanto, será testado as quatro diferentes arquiteturas de redes neurais convolucionais indicadas na Fig. 4 e denominados modelos 1, 2, 3 e 4, respectivamente. Todos os testes serão executados com a mesma divisão de dados para treinamento e teste. O treinamento de todos os modelos será realizado com a ferramenta Google Colab e a utilização de GPU para agilizar o processo.

1) CNN convencional: conv-->maxpool-->conv-->maxpool-->Densa-->Densa-->predição

2) CNN + dropout: conv-->maxpool-->conv-->maxpool-->Densa-->Dropout-->Densa-->predição

3) CNN + batch normalization: conv-->BN-->ReLu-->maxpool-->conv-->BN-->ReLu-->maxpool-->Densa-->Densa-->predição

4) CNN + Global average pooling: conv-->maxpool-->conv-->GAP-->Densa-->Densa-->predição

Figure 4. Arquiteturas de CNN sequenciais

Em seguida, serão também avaliados os desempenhos dos modelos pré-treinados InceptionV4, também conhecido como InceptionResNetV2 (Szegedy *et al.* [16]) e Xception (Chollet [17]), a Tab. 2 apresenta uma comparação entre o número de parâmetros e camadas dos modelos pré-treinados adotados. No desenvolvimento deste trabalho serão denominados modelos 5 e 6 respectivamente. Ambos terão uma camada de ativação do tipo *softmax*, o otimizador Adam e a loss do tipo *categorical crossentropy*, por ser um problema de classificação multiclasse. A taxa de apresendizado de todos os modelos adotada é de 0.0001 e os modelos sequencias são treinados com 200 épocas enquanto que para os pré-treinados o valor de 50 épocas foi considerado.

Tabela 1. Comparação entre os modelos pré-treinados

Modelo	**Parâmetros**	**Camadas**
InceptionResNetV2	55,873,736	572
Xception	22,910,480	126

Para o modelo Xception será também avaliado o desempenho com o otimizador Nadam (modelos 7 e 8), a diferença entre estes está no fato que o modelo 7 utiliza uma taxa de aprendizado com decaimento em platô (com o comando ReduceLROnPlateau (factor = 0.7, patience = 2)) e o modelo 8 adota uma taxa de aprendizado com decaimento exponencial (com o comando LearningRateScheduler (exponential_decay_fn)).

Em todos os modelos avaliados o desbalanceamento entre as classes será tratado de forma que a classe com menor número de amostras ganha mais peso e é penalizada de acordo com isso durante o treinamento. Além disso, a técnica de validação cruzada [18] será aplicada no modelo de classificação

que obtiver melhor acurácia, com 10 Folds (número de dobras) que é o mais usual, para selecionar os dados de teste e treinamento.

A avaliação do desempenho de cada um dos modelos descritos será realizada a partir de análises das matrizes de confusão, com estatísticas de avaliação como: Recall, Precisão, F1 e Acurácia. A seguir será feita uma breve explanação de cada um destes métodos de avaliação de desempenho.

Sabe-se que a acurácia para classificadores não é considerada a melhor métrica de desempenho, principalmente quando se analisa com conjuntos de dados desbalanceados, porem está será calculada em conjunto com outras métricas recomendadas para esse tipo de problema. A matriz de confusão é o método mais indicado para avaliar o desempenho de um modelo de classificação [18]. Em problemas de duas classes a matriz é construída da seguinte forma: cada linha em uma matriz de confusão representa uma classe real e cada coluna representa uma classe prevista. Para um problema de duas classes, a matriz será 2x2 (Fig. 5) e a primeira posição representa o número de verdadeiro positivos, ou seja, valores que são verdadeiros e foram classificados como tal, porém a posição primeira linha e segunda coluna representa o número de falsos positivos, valores que correspondem a classe negativo e foram classificados como positivos, e assim respectivamente, a posição da segunda linha e primeira coluna corresponde aos falsos negativos, ou seja, pertencem a classe.

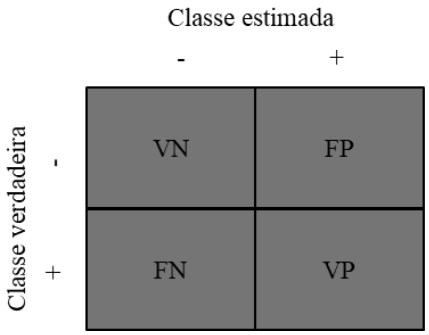

Figura 5. Matriz de confusão

Da matriz de confusão podem ser retiradas outras métricas como a acurácia das previsões positivas, também chamada de precisão do classificador, indicada na Eq. 1. Onde VP é o número de verdadeiros positivos e FP é o número de falsos positivos. A precisão é analisada em conjunto métrica revocação (Eq. 2), que pode ser entendida como a sensibilidade ou taxa de verdadeiros positivos, ou seja é a taxa de registros positivos que são corretamente classificadas. Onde FN é o número de falsos negativos. Na biblioteca *scikit learn* essas métricas são calculadas com a função precision_score (precisão) e recall_score (revocação).

$$precisão = \frac{VP}{VP+FP} \quad (1)$$

$$revocação = \frac{VP}{VP+FN} \quad (2)$$

Por fim, é comum combinar precisão e revocação em um único incide, chamado de pontuação F1, que é a média harmônica entre as duas métricas apresentadas anteriormente (Eq. 3). No *scikit-learn* essa métrica é calculada com o comando F1_score.

$$F_1 = \frac{2}{\frac{1}{precisão}+\frac{1}{revocação}} = \frac{VP}{VP+\frac{FN+FP}{2}} \quad (3)$$

4. Resultados

A seguir estão ilustrados os resultados de matriz de confusão e as tabelas com as métricas para cada um dos modelos de CNN em análise. E por fim uma comparação de todas as avaliações de desempenho calculadas para todos os modelos.

4.1 Modelo 1

Tabela 2. Métricas de avaliação para o Modelo 1

Classe	precisão	recall	f1-score
cardboard	0.88	0.75	0.81
glass	0.58	0.59	0.59
metal	0.65	0.45	0.53
paper	0.61	0.94	0.74
plastic	0.66	059	0.62
trash	0.43	0.12	0.19

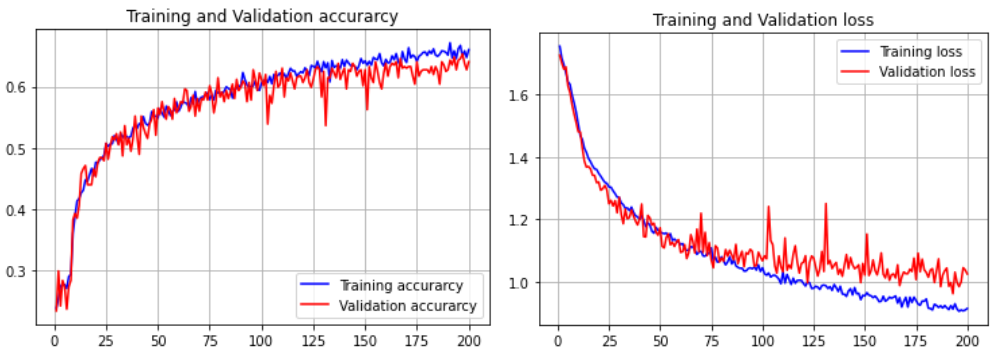

Figura 6. Evolução da acurácia e da perda ao longo das épocas

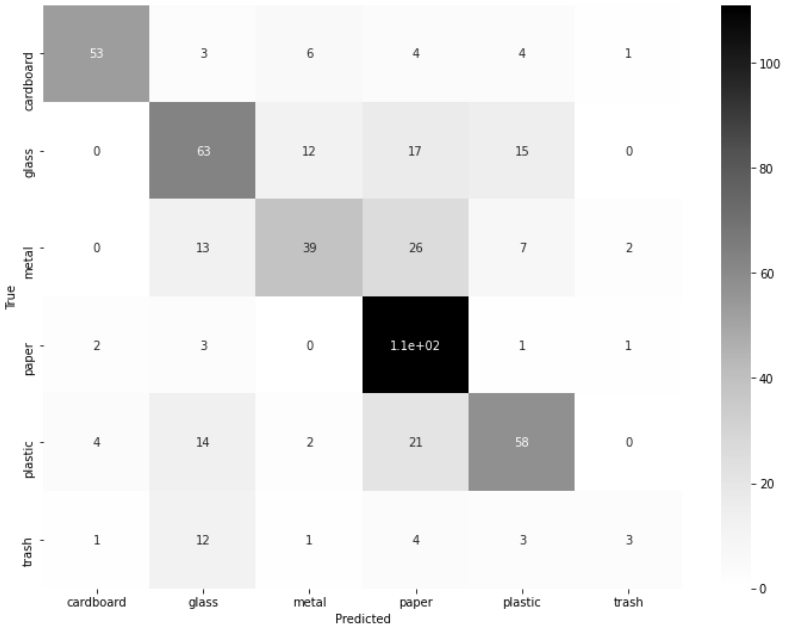

Figure 7. Matriz de Confusão (Modelo 1)

4.2 Modelo 2

Tabela 3. Métricas de avaliação para o Modelo 2

Classe	precisão	recall	f1-score
cardboard	0.92	0.76	0.83
glass	0.48	0.37	0.42
metal	0.66	0.45	0.53
paper	0.52	0.94	0.67
plastic	0.56	0.51	0.53
trash	0.00	0.00	0.00

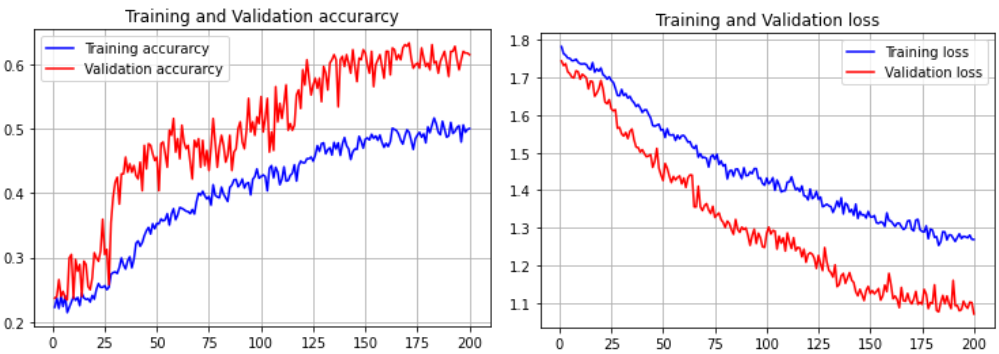

Figura 8. Evolução da acurácia e da perda ao longo das épocas

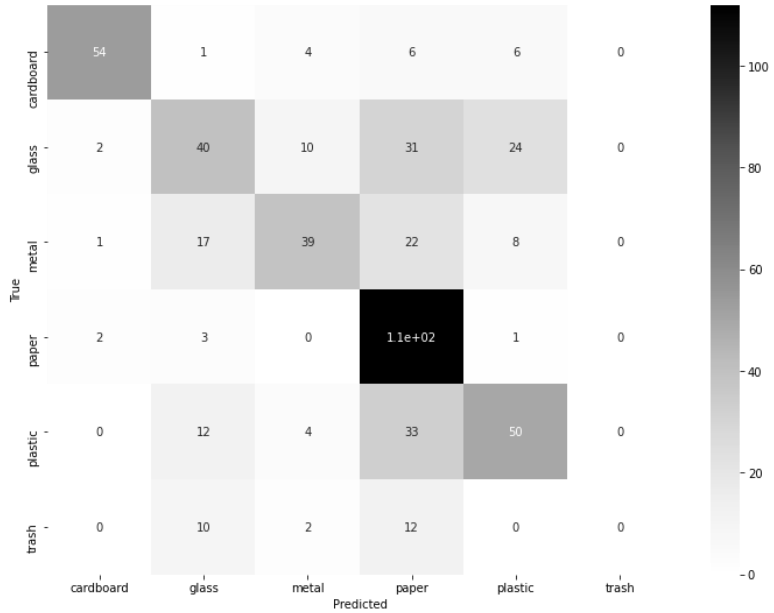

Figura 9. Matriz de Confusão (Modelo 2)

4.3 Modelo 3

Tabela 4. Métricas de avaliação para o Modelo 3

Classe	precisão	recall	f1-score
cardboard	0.68	0.87	0.77
glass	0.67	0.36	0.47
metal	0.64	0.61	0.62
paper	0.67	0.93	0.78
plastic	0.65	0.61	0.63
trash	0.58	0.46	0.51

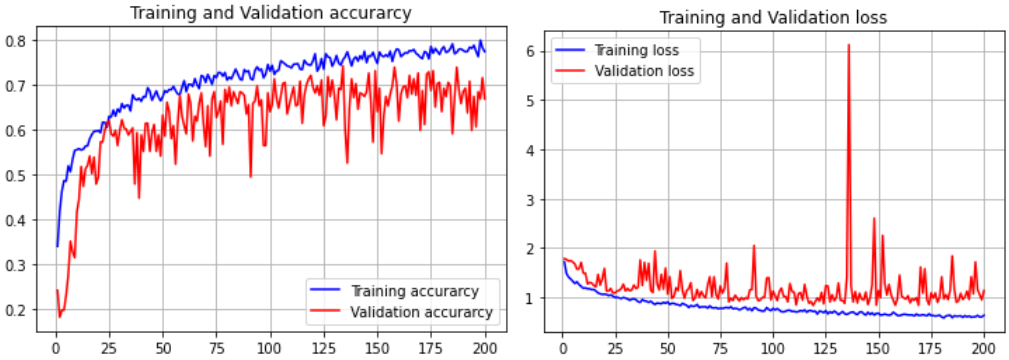

Figura 10. Evolução da acurácia e da perda ao longo das épocas

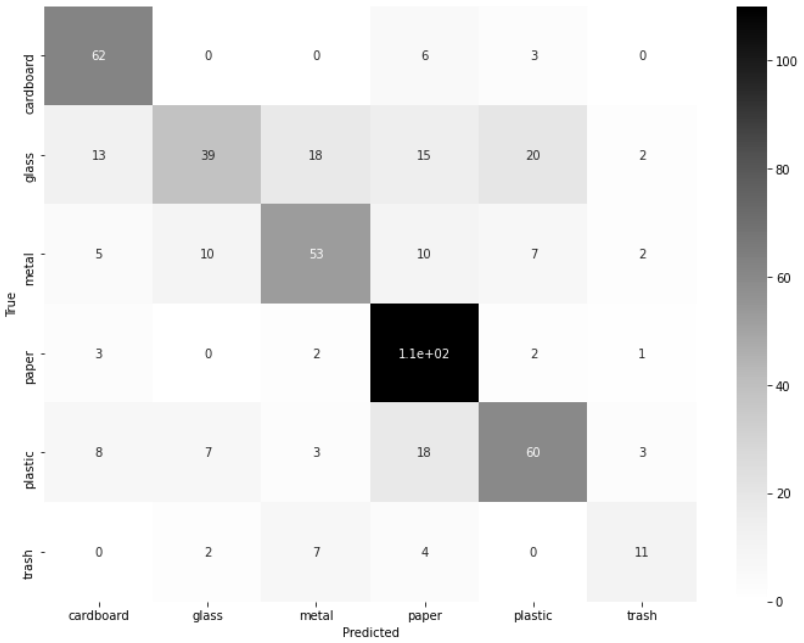

Figura 11. Matriz de Confusão (Modelo 3)

4.4 Modelo 4

Tabela 5. Métricas de avaliação para o Modelo 4

Classe	precisão	recall	f1-score
cardboard	0.82	0.79	0.81
glass	0.65	0.62	0.63
metal	0.70	0.40	0.51
paper	0.71	0.93	0.80
plastic	0.59	0.66	0.62
trash	0.52	0.46	0.49

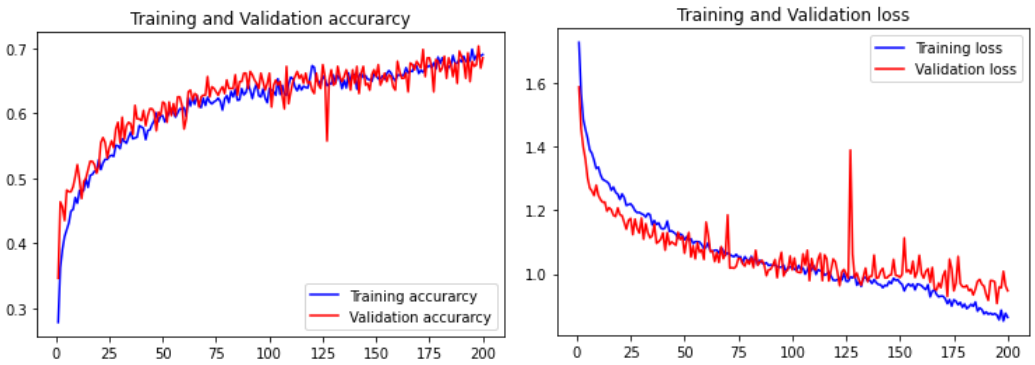

Figura 12. Evolução da acurácia e da perda ao longo das épocas

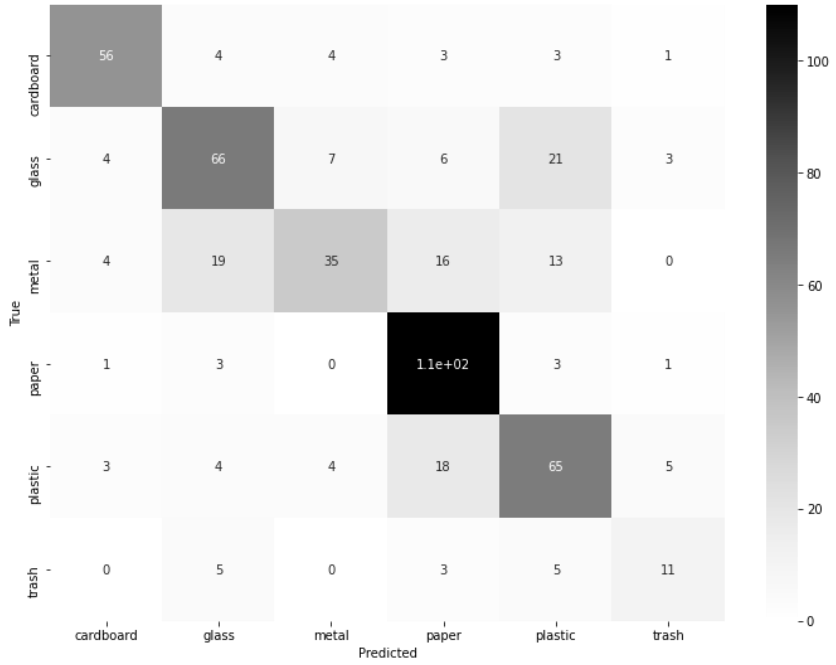

Figura 13. Matriz de Confusão (Modelo 4)

4.5 Modelo 5

Tabela 6. Métricas do modelo 5

Classe	precisão	recall	f1-score
cardboard	0.95	0.90	0.93
glass	0.97	0.78	0.86
metal	0.83	0.94	0.88
paper	0.90	0.95	0.92
plastic	0.83	0.91	0.86
trash	0.88	0.79	0.84

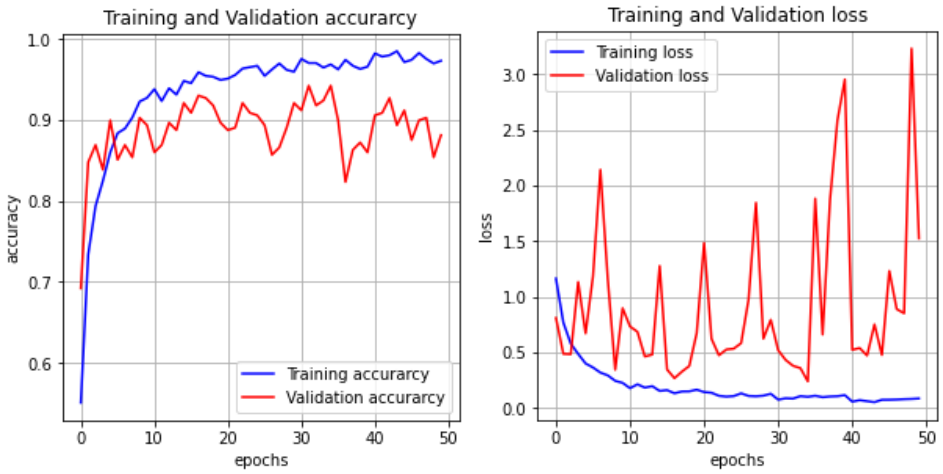

Figura 14. Evolução da acurácia e da perda ao longo das épocas

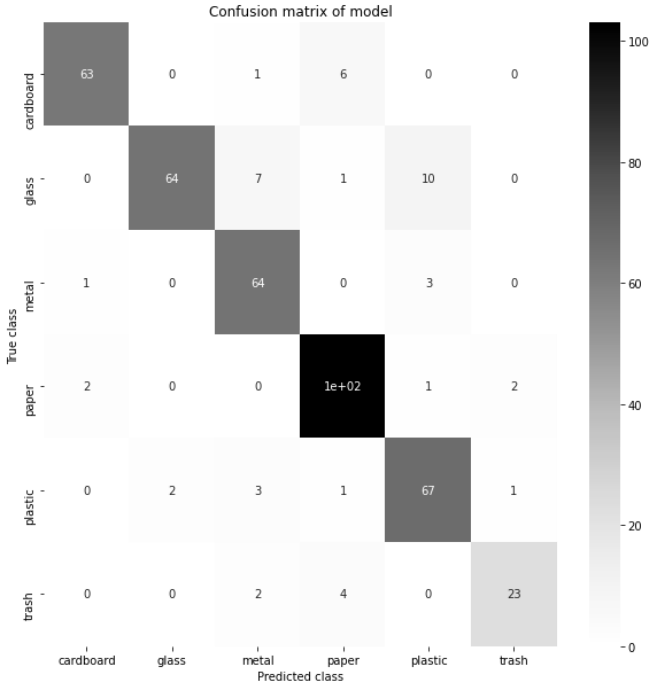

Figura 15. Matriz de Confusão (Modelo 5)

4.6 Modelo 6

Table 7. Métricas de avaliação para o Modelo 6

Classe	precisão	recall	f1-score
cardboard	0.81	0.96	0.88
glass	0.93	0.78	0.85
metal	0.74	0.96	0.83
paper	0.95	0.75	0.84
plastic	0.86	0.82	0.84

trash	0.80	0.97	0.88

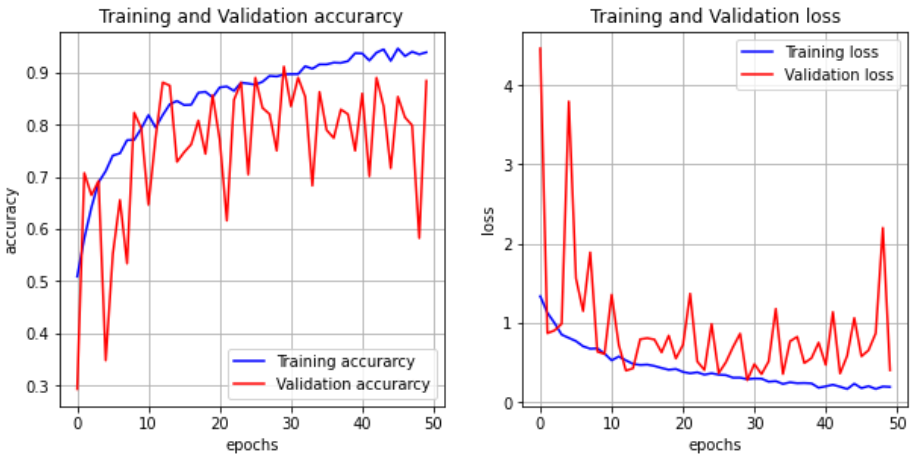

Figura 16. Evolução da acurácia e da perda ao longo das épocas

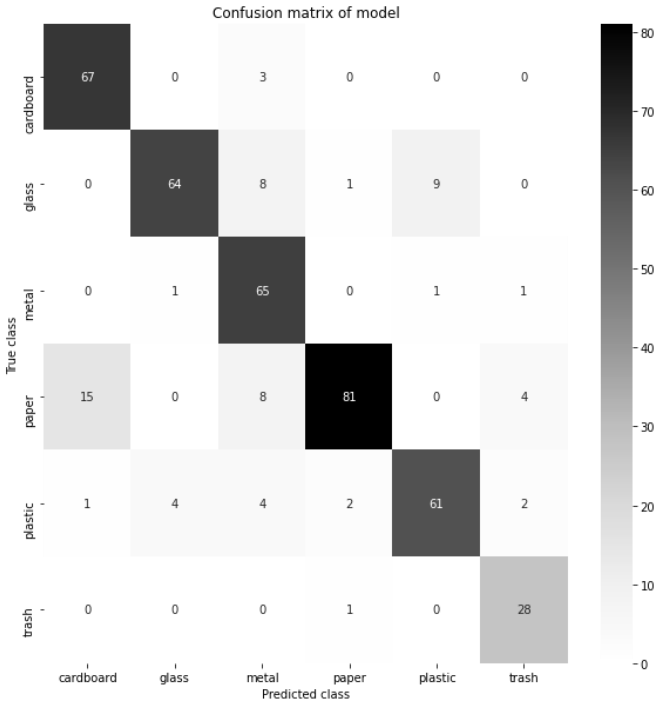

Figura 17. Matriz de Confusão (Modelo 6)

4.7 Modelo 7

Tabela 8. Métricas de avaliação para o Modelo 7

Classe	precisão	recall	f1-score
cardboard	0.94	0.97	0.96
glass	0.96	0.87	0.91
metal	0.92	0.99	0.95
paper	0.96	0.94	0.95

plastic	0.94	0.91	0.92
trash	0.80	0.97	0.88

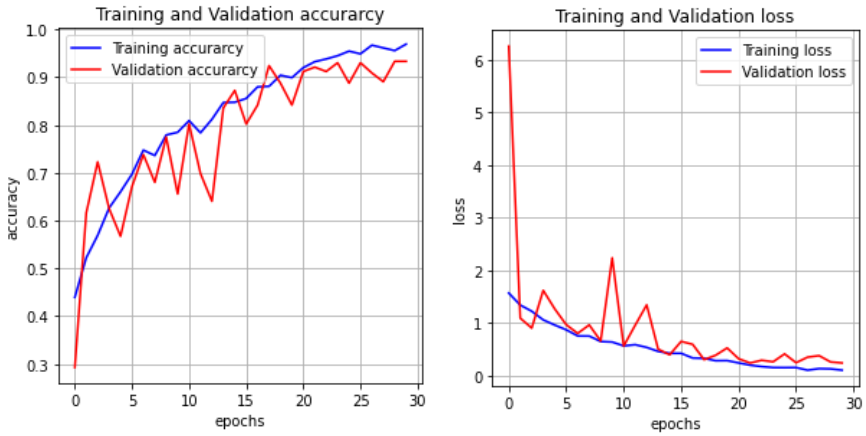

Figura 18. Evolução da acurácia e da perda ao longo das épocas

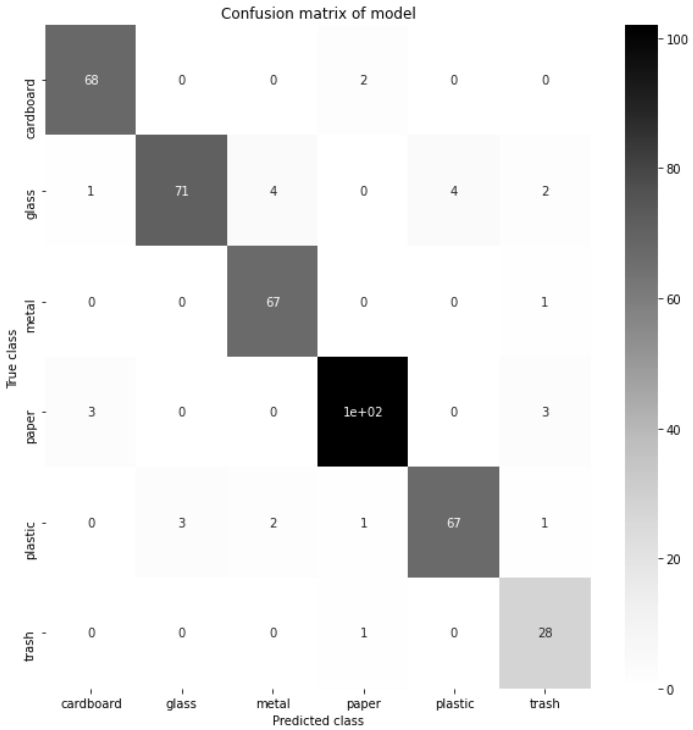

Figura 19. Matriz de Confusão (Modelo 7)

Como apontado na metodologia, o modelo 7 possui uma taxa de aprendizado com decaimento em platô, a Fig. 20 indica esse decaimento ao longo das épocas.

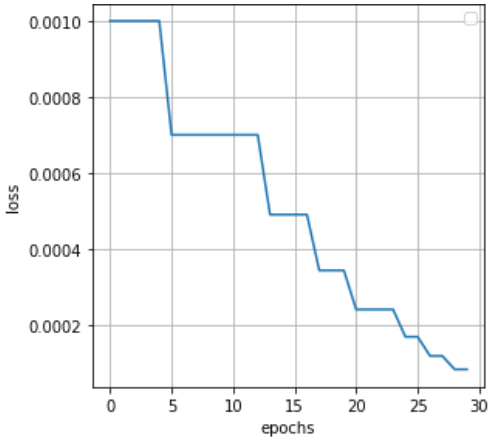

Figura 20. Decaimento da taxa de aprendizado

4.8 Modelo 8

Tabela 9. Métricas de avaliação para o Modelo 8

Classe	precisão	recall	f1-score
cardboard	0.97	0.94	0.96
glass	0.93	0.90	0.91
metal	0.92	0.96	0.94
paper	0.94	0.98	0.96
plastic	0.93	0.91	0.92
trash	0.93	0.86	0.89

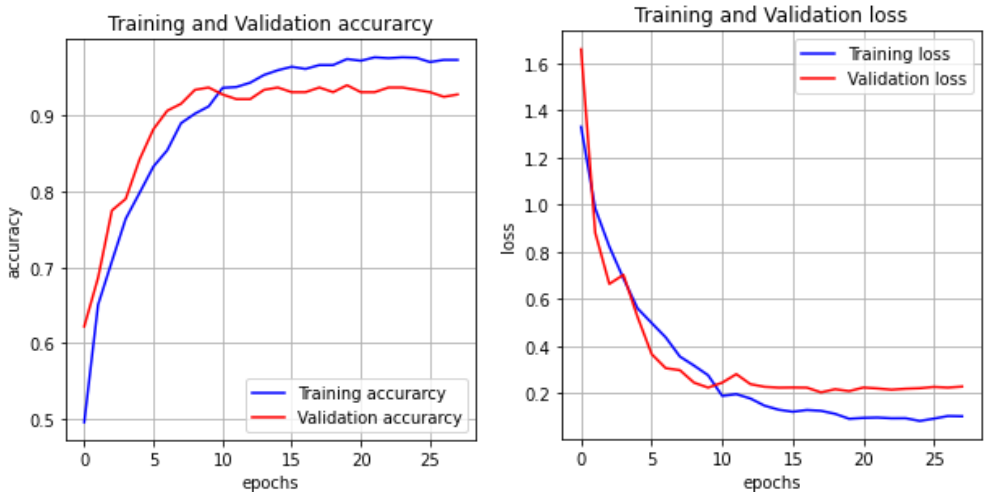

Figura 21. Evolução da acurácia e da perda ao longo das épocas

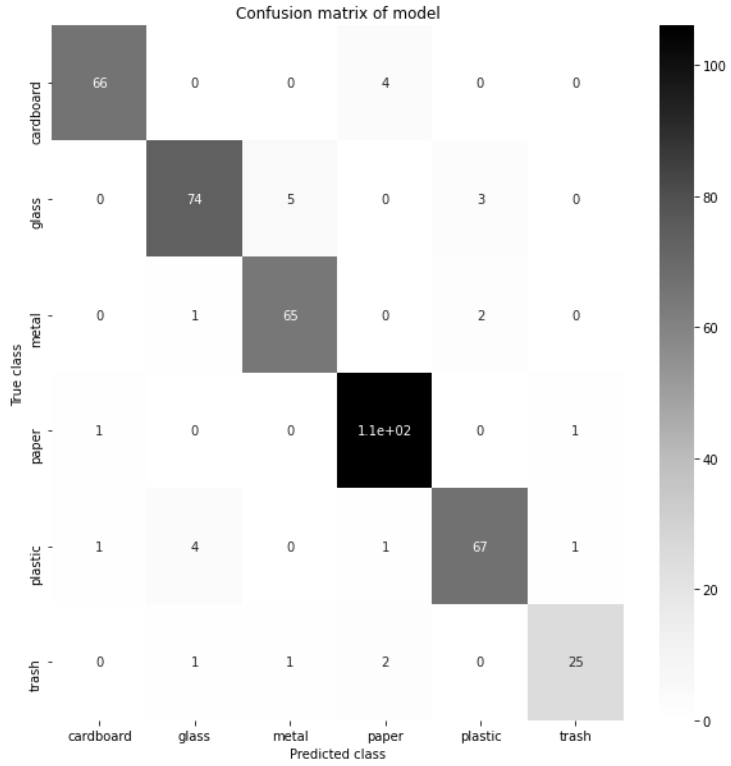

Figura 22. Matriz de Confusão (Modelo 8)

A Fig. 23 apresenta o decaimento da taxa de aprendizado com função exponencial, o que resultou em melhoras para o modelo.

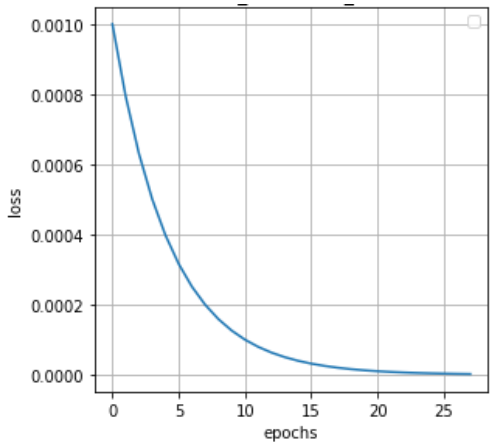

Figura 23. Decaimento da taxa de aprendizado

A Tab. 7 apresenta o resumo dos resultados de todos os modelos avaliados no trabalho. Observa-se que o modelo que obteve melhor performance foi o modelo 8.

Tabela 10. Resumo de resultados dos modelos CNN

Modelos Sequenciais			Modelos Pré-treinados		
Modelo	**Nº Épocas**	**Acurácia**	**Modelo**	**Nº Épocas**	**Acurácia**
1	200	0.6462	5	50	0.8909
2	200	0.5830	6	50	0. 8492
3	200	0.6620	7	50	0.93503
4	200	0.6778	8	50	0.94503

5. Conclusões

O presente trabalho conseguiu aplicar modelos de redes neurais convolucionais em um conjunto de dados composto por fotos de lixo, denominado como TrashNet, dessa forma foi possível prever ou classificar o tipo de lixo que uma imagem apresenta com até 95% de acurácia.

Com a execução de todos os modelos de classificação propostos neste trabalho pode-se afirmar que o modelo mais indicado para este conjunto de dados é o Modelo 8, que é o modelo pré-treinado Xception, com otimizador Nadam e decaimento da taxa de aprendizado exponencial, uma vez que este foi o modelo que apresentou maior acurácia e matriz de confusão com mais valores na diagonal princial, além das métricas precisão, recall e F1-score com valores mais próximos da unidade. Estes medem a capacidade de um modelo classificar corretamente um dado, que é o melhor índice de avaliação para modelos de classificação multiclasse. Cabe ressaltar que a complexidade do modelo não necessariamente está relacionada à um melhor resultado, isto foi comprovado com os melhores resultados do modelo 8 que é um modelo com metade do número de camadas que o modelo 5.

No presente trabalho dedicou-se uma certa cautela à etapa de pré-processamento com regularização das imagens e aumento do conjunto de dados, acredita-se que por isso os resultados dos modelos foram satisfatórios. Pode-se concluir que o pré-processamento tem papel fundamental na qualidade final dos resultados. Além disso, o cuidado com o tratamento do conjunto de dados desbalanceados, como o treinamento dos modelos de forma que esse desbalanceamento seja convertido, resultou em resultados mais confiáveis.

Os resultados encontrados neste trabalho se mostraram melhores que os do trabalho de Ozkaya e Seyfi [7] e Aral *et al*. [8], acredita-se que essa melhora foi devido ao pré-processamento adotado. Os modelos de classificação ainda poderiam ser melhorados com estudos na busca dos parâmetros para cada modelo com a técnica de Grid Search, essa recomendação é dada para trabalhos futuros.

Referências

[1] YANG, Mindy; THUNG, Gary. Classification of trash for recyclability status. CS229 Project Report, v. 2016, 2016.

[2] TrashNet LINK: https://github.com/garythung/trashnet

[3] T. Guo, J. Dong, H. Li and Y. Gao, "Simple convolutional neural network on image classification," 2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA), Beijing, 2017, pp. 721-724, doi: 10.1109/ICBDA.2017.8078730.

[4] LecunY, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11):2278-2324.

[5] G. Thung, “Trashnet,” GitHub repository, 2016.

[6] BIRCANOĞLU, Cenk et al. Recyclenet: Intelligent waste sorting using deep neural networks. In: 2018 Innovations in Intelligent Systems and Applications (INISTA). IEEE, 2018. p. 1-7.

[7] ARAL, Rahmi Arda et al. Classification of trashnet dataset based on deep learning models. In: 2018 IEEE International Conference on Big Data (Big Data). IEEE, 2018. p. 2058-2062.

[8] OZKAYA, Umut; SEYFI, Levent. Fine-tuning models comparisons on garbage classification for recyclability. arXiv preprint arXiv:1908.04393, 2019.

[9] WANG, Yuheng et al. Recyclable Waste Identification Using CNN Image Recognition and Gaussian Clustering. arXiv preprint arXiv:2011.01353, 2020.

[10] NumPy v1.19 Manual Documentation. Disponível em: <https://numpy.org/doc/stable/reference/>. Acesso em: 17 de ago. de 2020.

[11] Pandas v1.11 Documentation. Disponível em: <https://pandas.pydata.org/docs/>. Acesso em: 17 de ago. de 2020.

[12] Matplotlib: Visualization with Python v3.3.1. Disponível em: <https://matplotlib.org/contents.html>. Acesso em: 17 de ago. de 2020.

[13] Seaborn: Statistical data visualization. Disponível em: <https://seaborn.pydata.org/tutorial.html>. Acesso em: 17 de ago. de 2020.

[14] ScikitLearn: Machine Learning in Python v0.23.2. Disponível em: <https://scikit-learn.org/stable/user_guide.html>. Acesso em: 18 de ago. de 2020.

[15] ABADI, Martín et al. Tensorflow: A system for large-scale machine learning. In: 12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16). 2016. p. 265-283.

[16] SZEGEDY, Christian et al. Inception-v4, inception-resnet and the impact of residual connections on learning. arXiv preprint arXiv:1602.07261, 2016.

[17] CHOLLET, François. Xception: Deep learning with depthwise separable convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2017. p. 1251-1258.

[18] Géron, A. Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: LecunY, Bottou