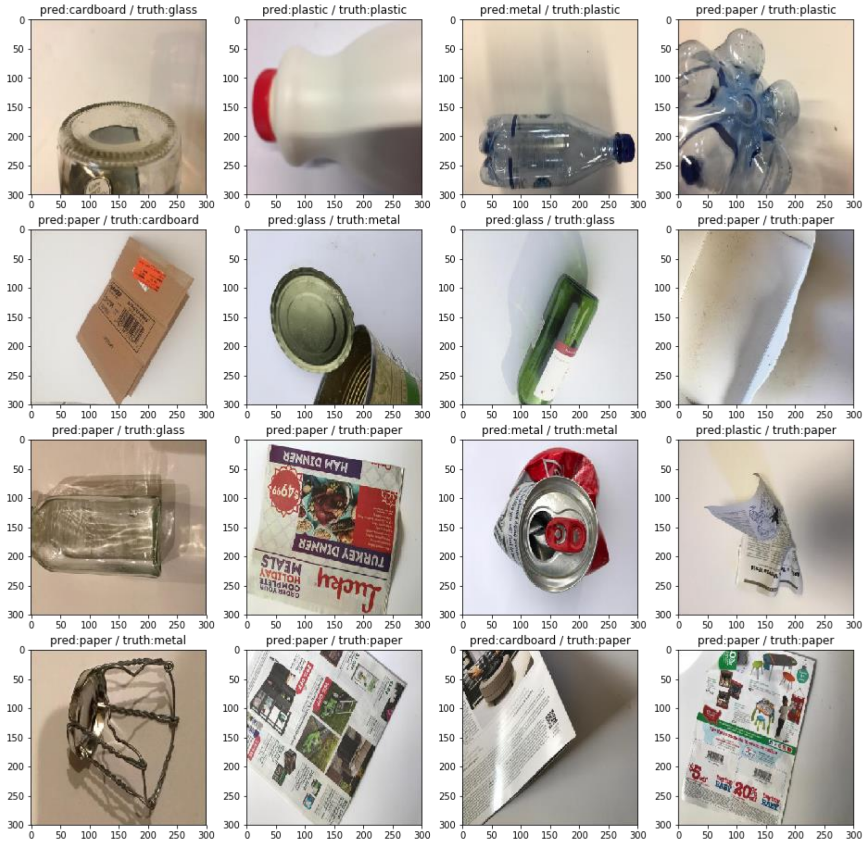
USO DE DEEP LEARNING PARA CLASSIFICAÇÃO DE IMAGENS DE LIXO



Amanda Isabela de Campos

COPPE UFRJ

Sumário

1. Introdução

Descrição do Problema:

Classificação de imagens

6 classes desbalanceadas

Pesquisa Bibliográfica

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.

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.

2. Tecnologia

Apresentação / Visualização de Dados

2527 imagens, com seis classes

vidro: 501;

papel: 594;

papelão: 403;

plástico: 482;

metal: 410 e

lixo: 137.

Papelão (*Cardboard*)

Vidro (*Glass*)

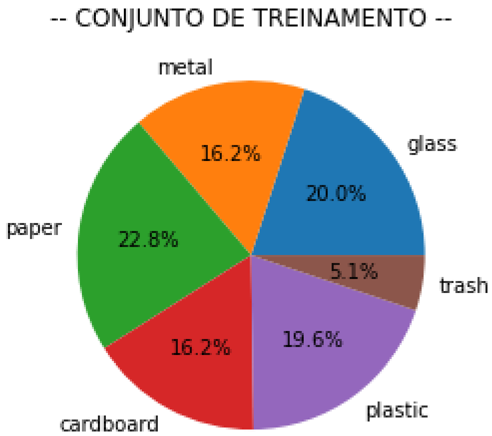
Metal (*Metal*)

Papel (*Paper*)

Plástico (*Plastic*)

Lixo (*Trash*)

3. Metodologia



Pré-processamento

Resize (150x150)

Treino/ Teste/ Validação (70/15/15)

Codificar valores de classe como números inteiros (LabelEncoder) e converter números inteiros em variáveis dummy (One-hot-encoded)

ImageDataGenerator

Normalizar dados de treinamento: rescale=1./255

ImageAugmentation: rotation_range = 40, width_shift_range = 0.2, height_shift_range = 0.2, shear_range = 0.2, zoom_range = 0.2,

horizontal_flip = True, vertical_flip = True

3. Metodologia

Modelos CNN sequencial

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

3. Metodologia

Modelos pré-treinados

GPU colab

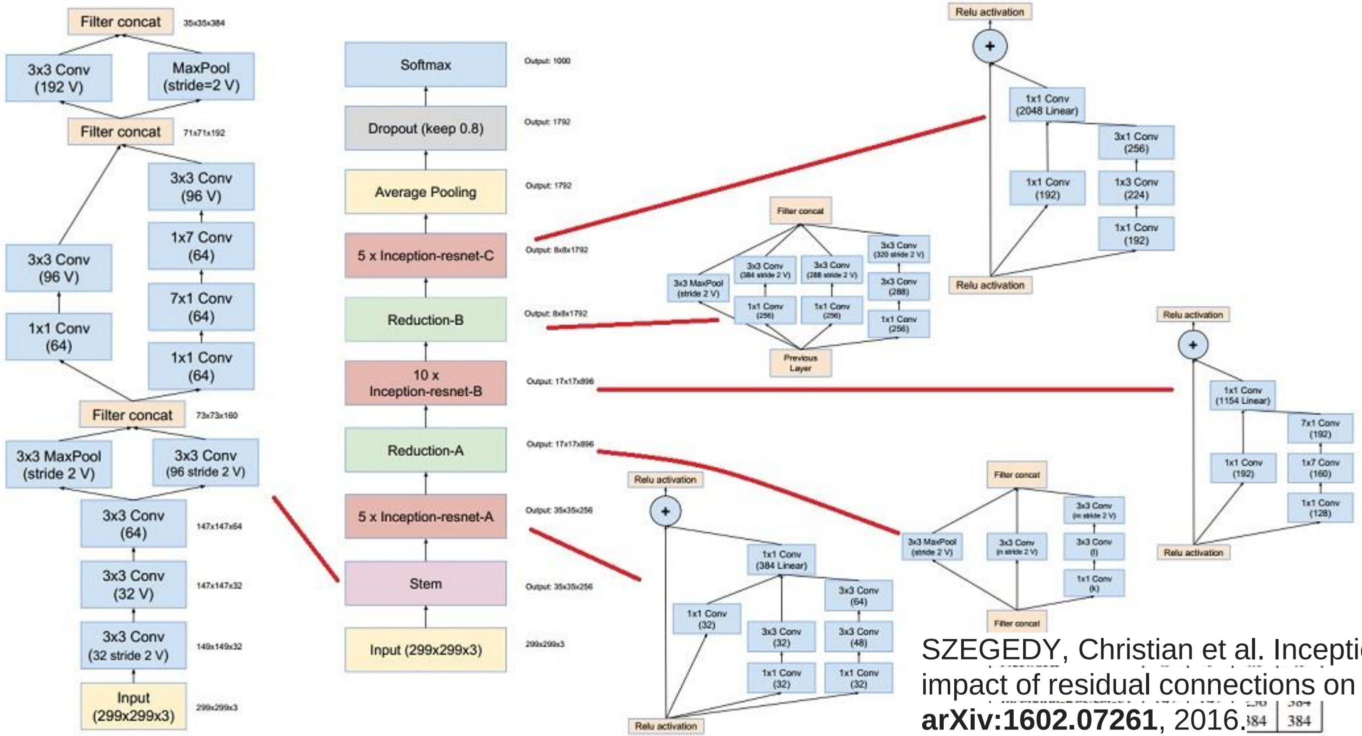
InceptionResNetV2 e Xception

```
output = keras.layers.Dense(6, activation="softmax")(avg)
model.compile(optimizer=optimizers.Adam(LR), loss='categorical_crossentropy', metrics=['acc'])
```

3. Metodologia

Modelos pré-treinados

InceptionResNetV2

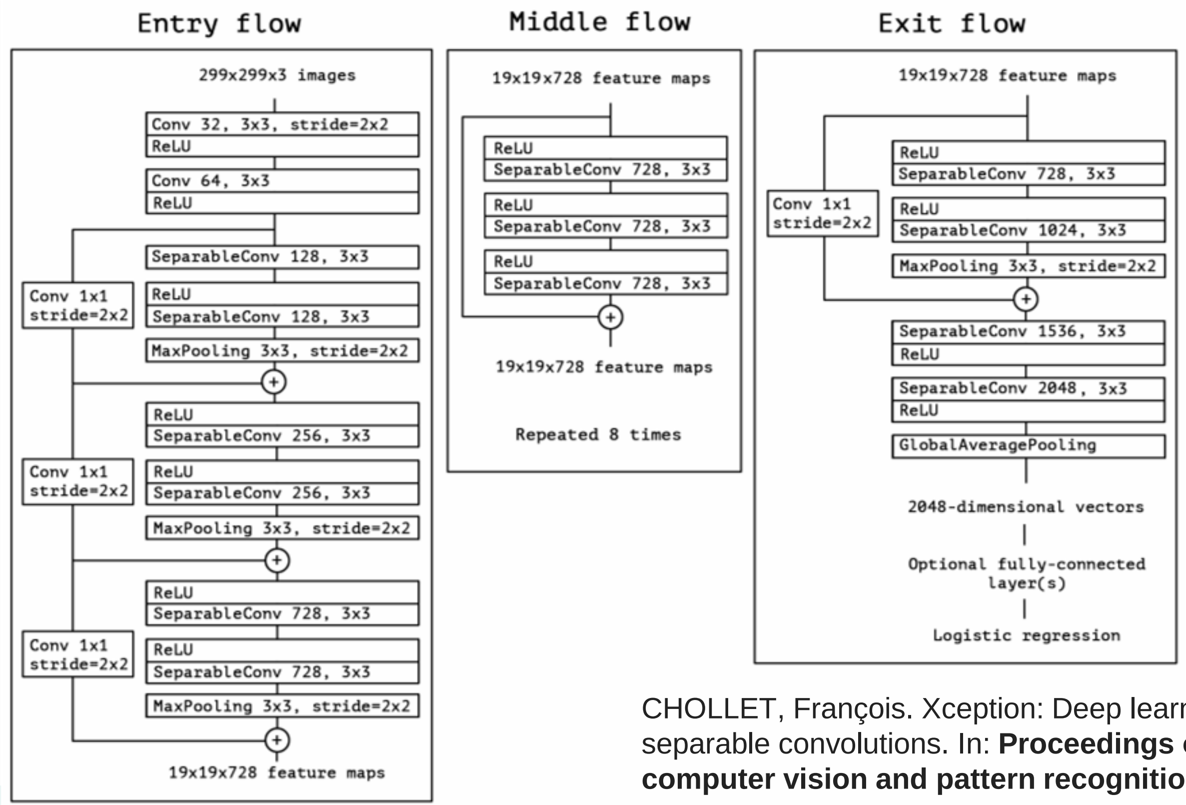


SZEGEDY, Christian et al. Inception-v4, inception-resnet and the impact of residual connections on learning. **arXiv preprint arXiv:1602.07261**, 2016.

3. Metodologia

Modelos pré-treinados

Xception



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.

3. Metodologia

Modelos pré-treinados

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

The class with the lowest number of samples gains more weight and is penalized accordingly during the training.

```
class_weights = []
total_samples = train_generator.samples
total_classes = len(train_generator.class_indices)
for ele in train_counts:
    result = round(total_samples / (total_classes * ele),2)
    class_weights.append(result)
print(dict(zip(labels,class_weights)))

class_weights = dict(zip(train_generator.class_indices.values(),class_weights))
```

```
{'glass': 0.83, 'metal': 1.03, 'paper': 0.73, 'cardboard': 1.03, 'plastic': 0.85, 'trash': 3.24}
```

Matriz de confusão

Métricas: acurácia, precisão, recall e F1-score

4. Resultados

CNN

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

Modelo	Nº Épocas	Acurácia
1	200	0.6462
2	200	0.5830
3	200	0.6620
4	200	0.6778

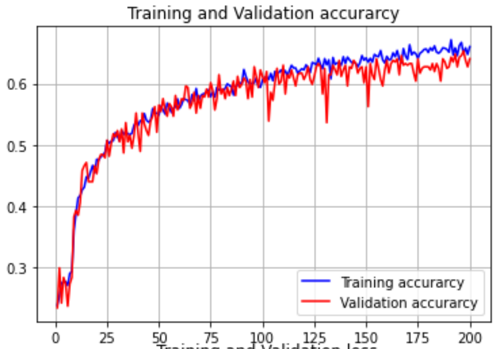
4. Resultados

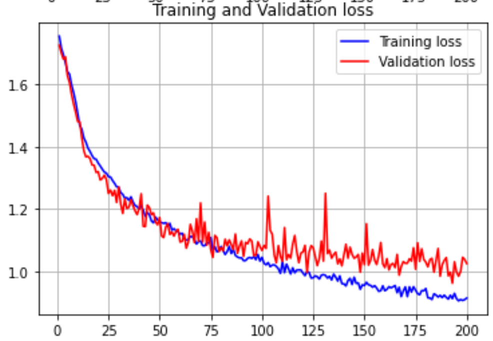
CNN

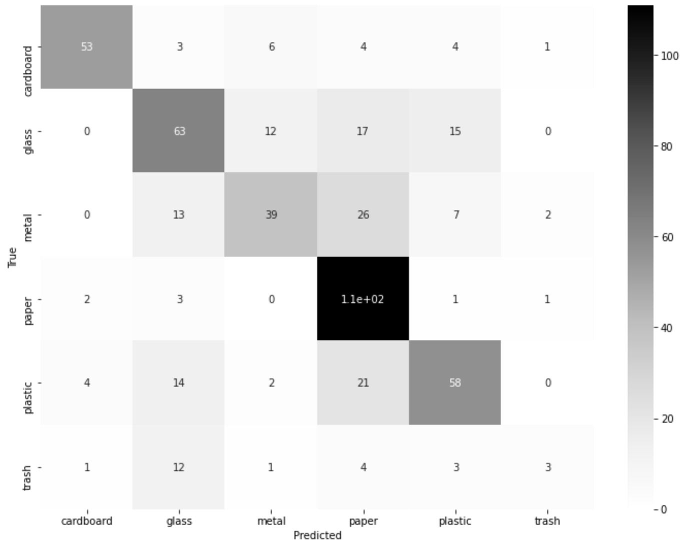
1) CNN convencional: conv-->maxpool-->conv-->maxpool-->Densa-->Densa-->predição

Test accuracy rate: 0.6462

Classe	precision	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







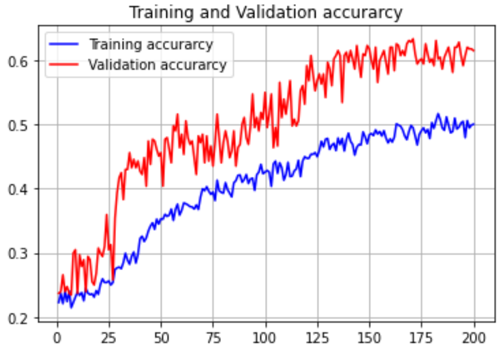
4. Resultados

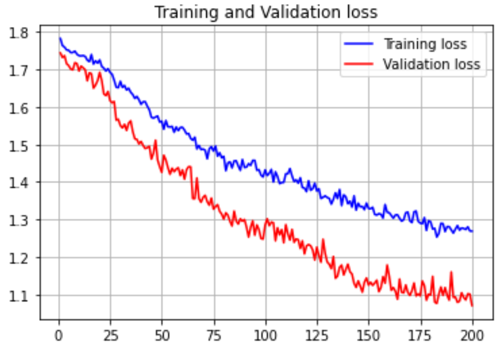
CNN

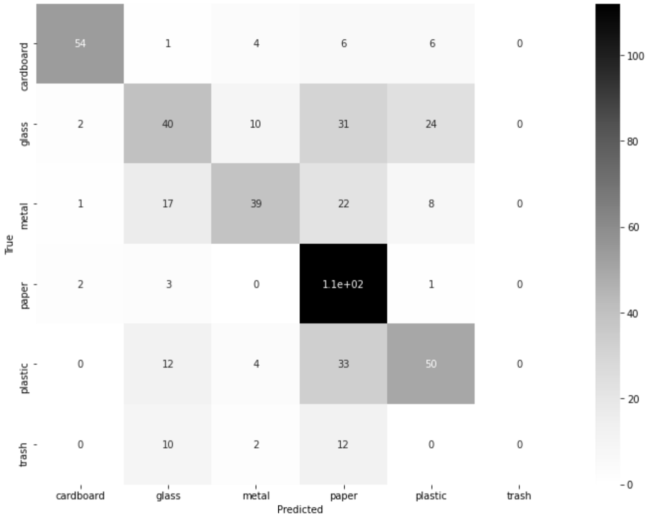
2) CNN + dropout: conv-->maxpool-->conv-->maxpool-->Densa-->Dropout-->Densa-->predição

Test accuracy rate: 0.5830

Classe	precision	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







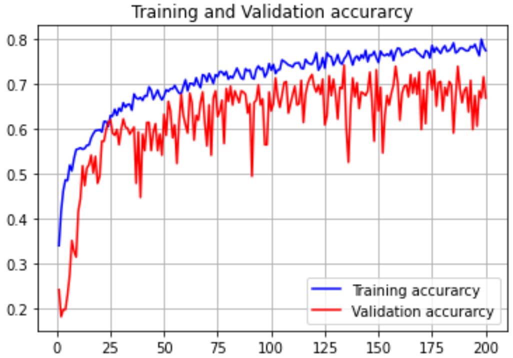
4. Resultados

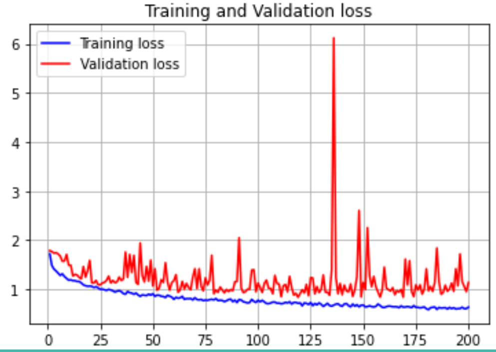
CNN

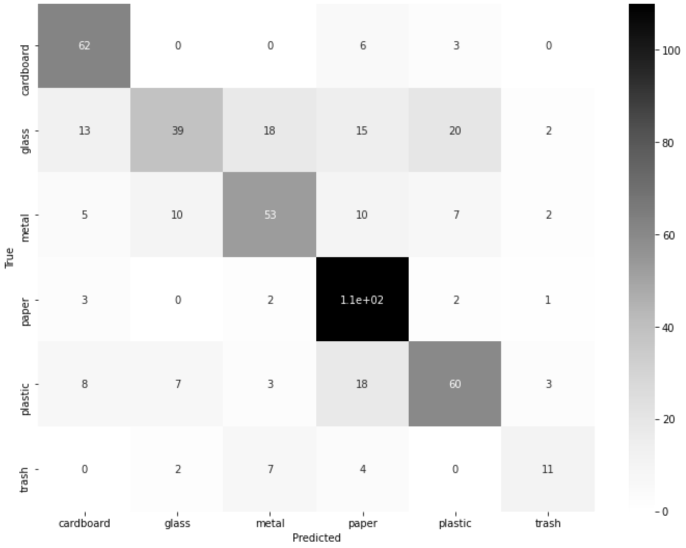
3) CNN + batch normalization: conv-->BN-->ReLu-->maxpool-->conv-->BN-->ReLu-->maxpool-->Densa-->Densa-->predição

Test accuracy rate: 0.6620

Classe	precision	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







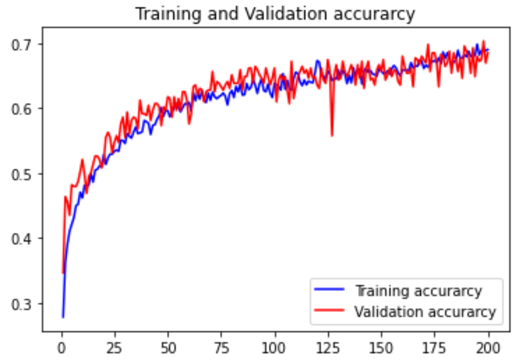
4. Resultados

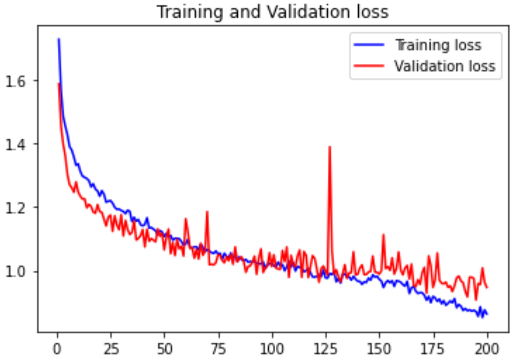
CNN

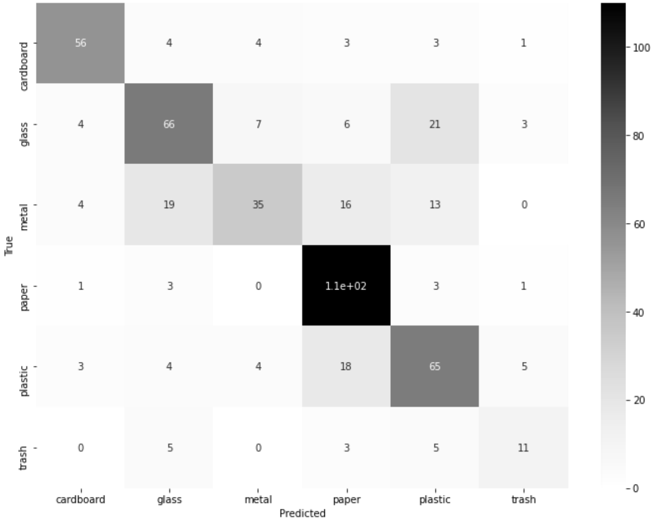
4) CNN + Global average pooling: conv-->maxpool-->conv-->GAP-->Densa-->Densa-->predição

Test accuracy rate: 0.6778

Classe	precision	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







4. Resultados

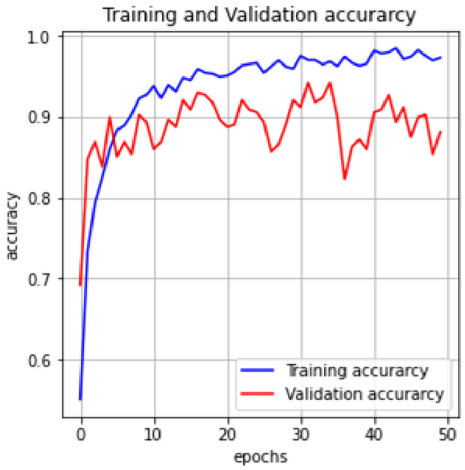
Modelos pré-treinados

5) Modelo InceptionResNetV2

optimizers.adam(0.0001)   epochs = 50

Test accuracy rate: 0. 89095

Classe	precision	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



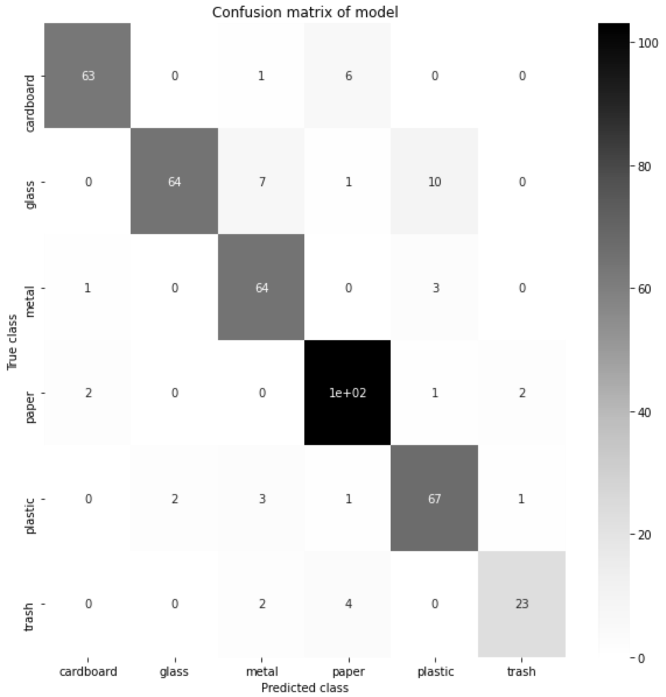


Figura 1. Matriz de Confusão

4. Resultados

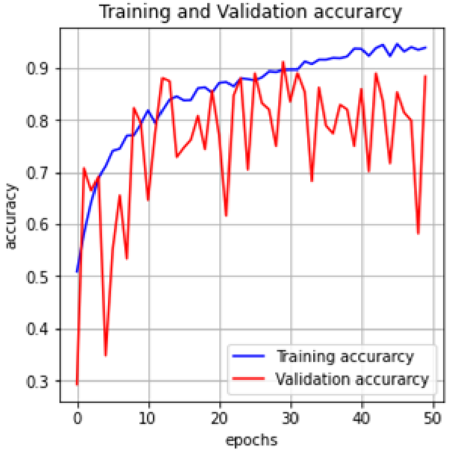
Modelos pré-treinados

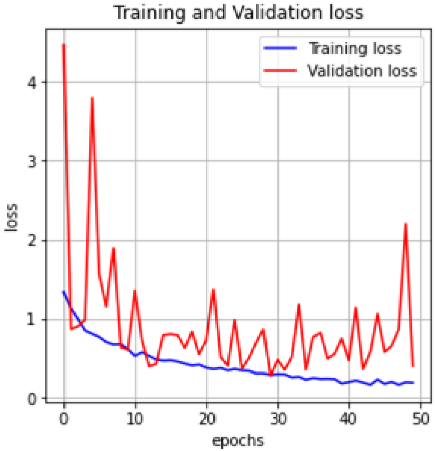
6) Modelo Xception

optimizers.adam(0.001)   epochs = 50

Test accuracy rate: 0. 84918

Classe	precision	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





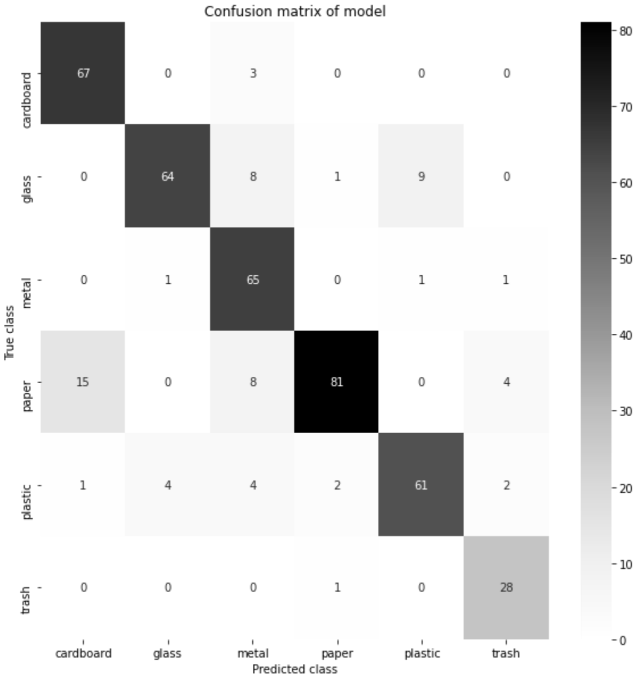


Figura 1. Matriz de Confusão

4. Resultados

Modelos pré-treinados

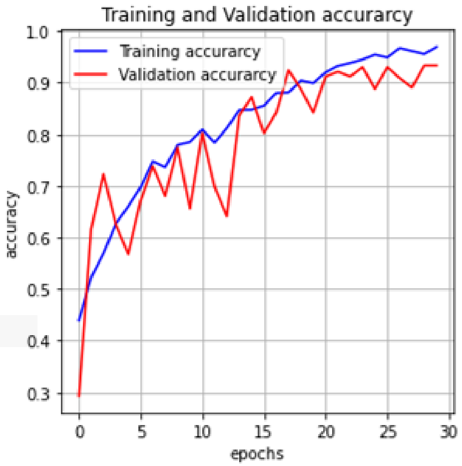
7) Modelo Xception

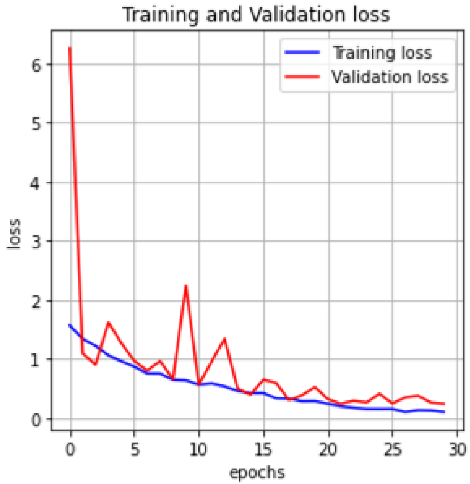
optimizers.Nadam(lr=0.001, beta_1=0.9, beta_2=0.999)

epochs = 30

Test accuracy rate: 0.93503

Classe	precision	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





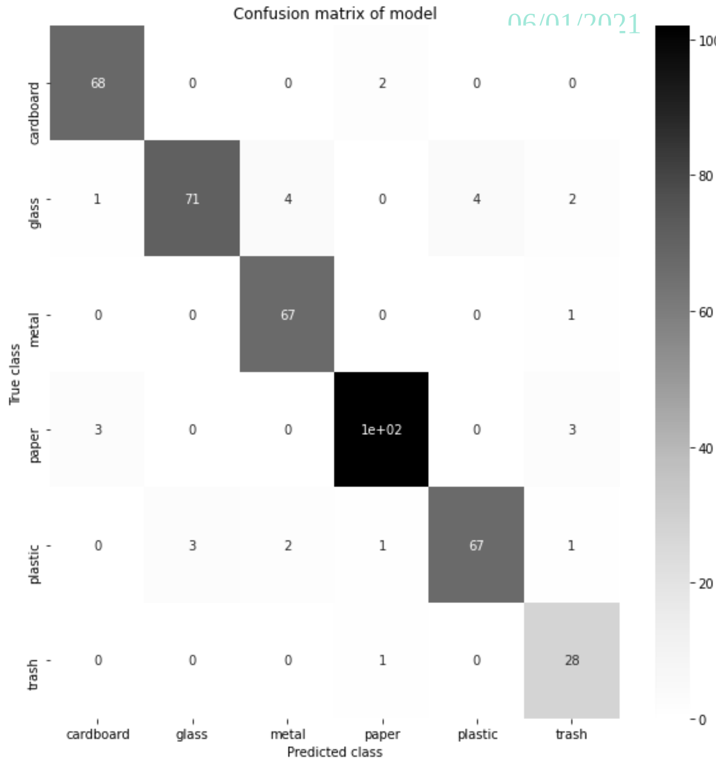


Figura 1. Matriz de Confusão

4. Resultados

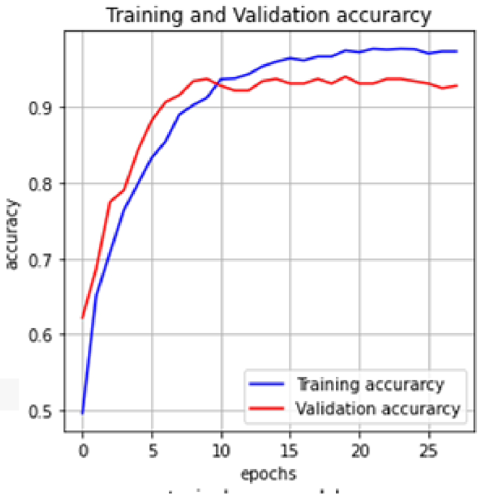
Modelos pré-treinados

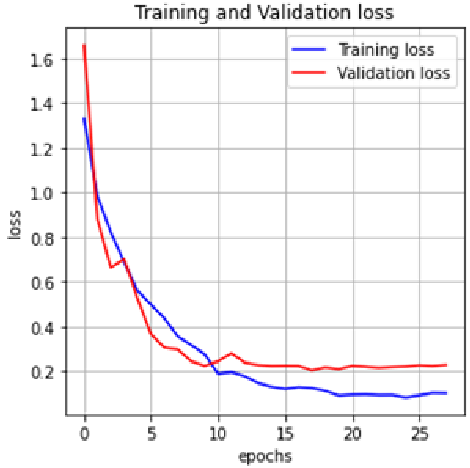
8) Modelo Xception

optimizers.Nadam(lr=0.001, beta_1=0.9, beta_2=0.999)
epochs = 50

Test accuracy rate: 0.94503

Classe	precision	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





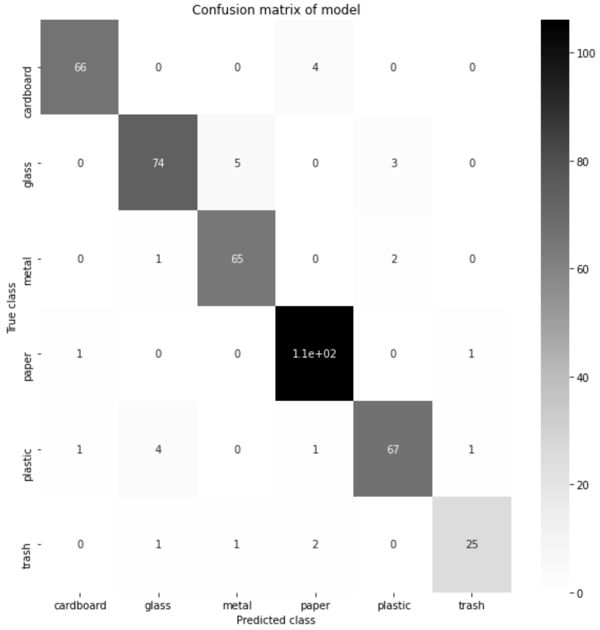
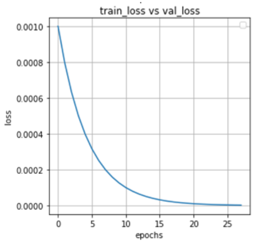
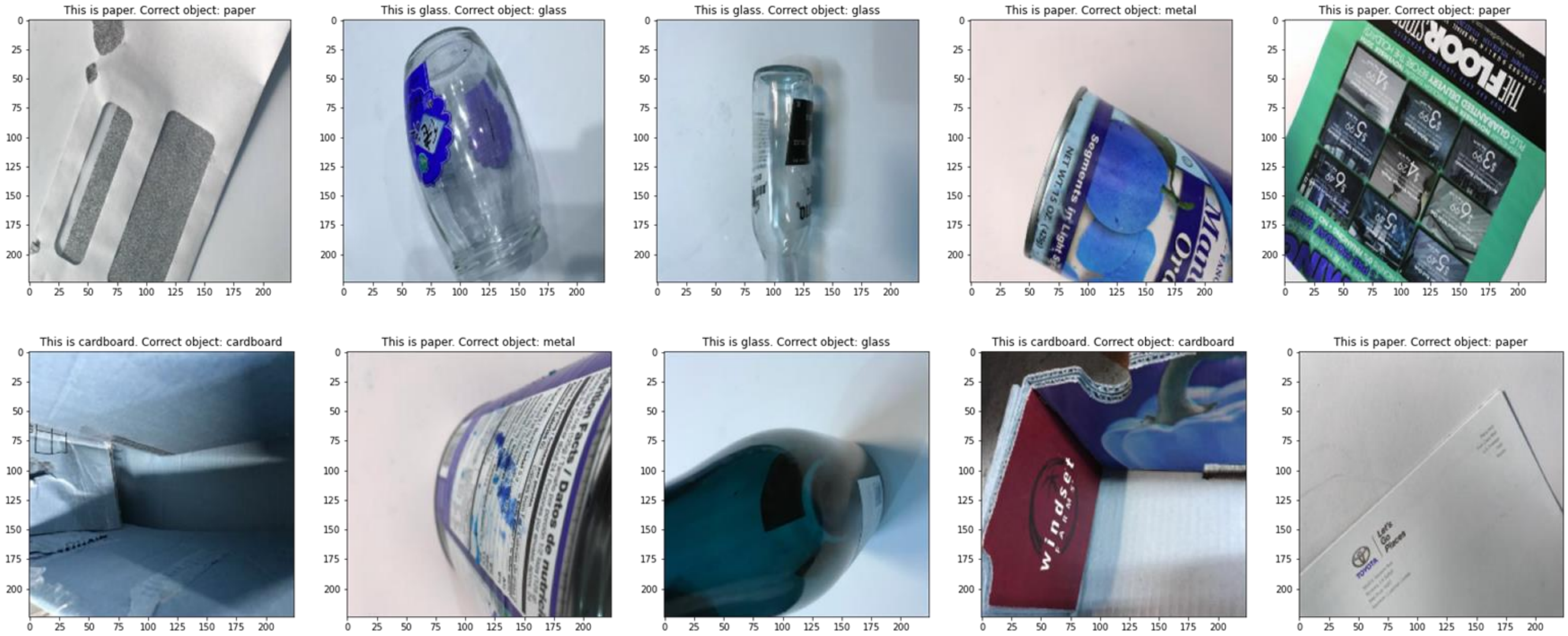


Figura 1. Matriz de Confusão



4. Resultados

Modelos pré-treinados



5. Conclusões

- O pré-processamento tem papel fundamental na qualidade final dos resultados;
- Conjunto de dados desbalanceados necessitam de um treinamento que considera o peso de cada classe;
- O modelo indicado para este conjunto de dados é o Modelo Xception ;
- A complexidade do modelo não necessariamente está relacionada à um melhor resultado;
- Recomenda-se um estudo mais aprofundado na busca dos parâmetros para cada modelo (*Grid Search*).

Referências

Aral, Rahmi Arda, et al. "Classification of trashnet dataset based on deep learning models." *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 2018.

Ozkaya, Umut, and Levent Seyfi. "Fine-tuning models comparisons on garbage classification for recyclability." *arXiv preprint arXiv:1908.04393* (2019).

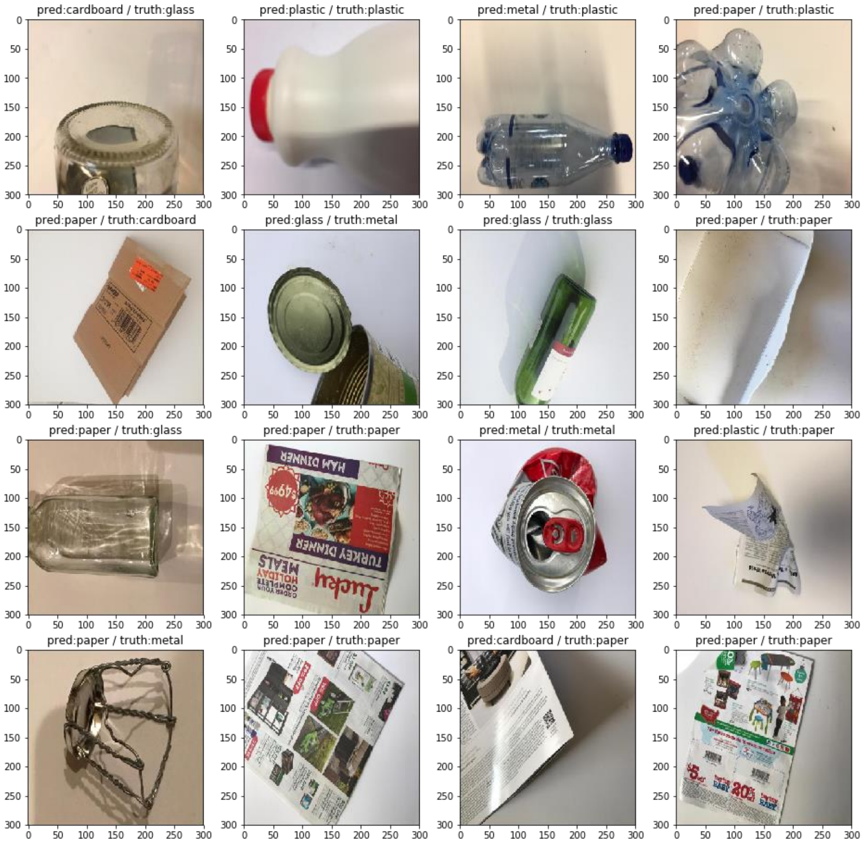
G. Thung, “Trashnet,” GitHub repository, 2016.

Thung, Gary and Mingxiang Yang. “Classification of Trash for Recyclability Status.” (2016).

Bircanoglu, C., Atay, M., Beser, F., Genc, O., & Kizrak, M. A. (2018). RecycleNet: Intelligent Waste Sorting Using Deep Neural Networks. 2018 Innovations in Intelligent Systems and Applications (INISTA). doi:10.1109/inista.2018.8466276

USO DE DEEP LEARNING PARA CLASSIFICAÇÃO DE IMAGENS DE LIXO

Obrigado!



Amanda Isabela de Campos

COPPE UFRJ