



Relatório do Trabalho Prático

Curitiba, 07 de Outubro de 2019.

Título do projeto: Classificação de Imagens

Equipe: Juliana Carolina Soldera da Silva

1. Introdução

Técnicas de classificação são encontradas em diversas funcionalidades presentes no dia-a-dia atual, como filtragem de spam, detecção de idioma, pesquisa por documentos semelhantes, análise de sentimentos, reconhecimento de caracteres e números manuscritos, detecção de fraude, entre outras. Classificar consiste em atribuir classes com base em características rotuladas previamente.

A etapa responsável pela extração de informações úteis é denominada extração de características, ou, extração de atributos. No caso de muitos dados, a ideia é que sejam selecionados os atributos que representem a parte relevante da informação.

Um classificador consiste em um modelo de aprendizado cuja saída é discreta, onde o objetivo é identificar características e rotular saídas das mesmas dentro de um contexto.

2. Identificação do problema

Criar e avaliar modelos de predição para a classificação de imagens usando os indutores aprendidos durante as aulas, ou a combinação destes em ensembles homogêneos ou heterogêneos.

3. Descrição dos experimentos

A base de imagens fornecida para criação dos modelos possui 10 classes e 1.000 imagens, sendo 100 exemplos de cada classe. As imagens estão organizadas em pastas

com o nome das classes: 'cavalos', 'comida', 'dino', 'elefante', 'flores', 'humanos', 'montanhas', 'obras', 'onibus' e 'praia'.

Através de um script fornecido em aula, foi realizada a extração de características das imagens. O script desenvolvido em *Python* usa as bibliotecas *OpenCV*, *Skimage* e *Keras* para realizar a extração de características a partir das imagens fornecidas.

Duas abordagens são consideradas para representação das imagens. Uma formada por *handcrafted features*, ou seja, um vetor composto por descritores de cor (histogramas RGB), forma (HOG, Histogramas de Orientação do Gradiente) e textura (LBP, *Local Binary Pattern*). A outra abordagem de representação utiliza a extração de características via rede convolucional (CNN, denominada *InceptionV3*) pré-treinada na base de imagens *ImageNet*.

Como saída do script, obtém-se três arquivos: *y.csv*, contendo o rótulo de cada imagem representado como um inteiro de 0 a 9, *X.csv*, arquivo com as *handcrafted features*, contendo 3.387 valores para cada entrada e *X_deep.csv*, contendo as *deep features* extraídas usando a CNN já treinada em outra base maior denominada *ImageNet*, com um total de 2.048 valores para cada imagem.

A partir das características obtidas, são treinados classificadores distintos (árvore de decisão, naive bayes, k-vizinhos mais próximos, redes neurais, máquina de vetor de suporte, etc) e, conseqüentemente, abordadas diferentes técnicas, a fim de avaliar suas respectivas saídas e verificar qual dos classificadores se mostra mais eficiente perante o problema proposto, além de avaliar qual das abordagens de extração de características contribui para o melhor resultado.

4. Resultados do processo

a. Taxa de acerto dos modelos avaliados:

- Decision Tree:
Classification accuracy - handcrafted: 0.67
Classification accuracy - deep features: 0.7733333333333333
- Naive Bayes:
Classification accuracy NaiveBayes - handcrafted: 0.8433333333333334
Classification accuracy NaiveBayes - deep features: 0.9433333333333334
- KNN:
Classification accuracy KNN - handcrafted: 0.3966666666666667
Classification accuracy KNN - deep features: 0.97
- MLP:

Classification accuracy MLP - handcrafted: 0.2

Classification accuracy MLP - deep features:0.97

- SVM:

Classification accuracy SVM - handcrafted: 0.71

Classification accuracy SVM - deep features: 0.9833333333333333

- Logistic Regression:

Classification accuracy LogisticRegression - handcrafted: 0.6633333333333333

Classification accuracy LogisticRegression – deep features: 0.9866666666666666

- RF:

Classification accuracy RF - handcraft: 0.8566666666666667

Classification accuracy RF - deep feaures: 0.9733333333333334

b. Comparação das duas abordagens de representação:

Analisando a taxa de acerto dos classificadores baseadas nas duas abordagens propostas, observou-se que o desempenho dos mesmos se mostrou melhor através da abordagem *deep features*. As figuras 1 e 2 representam as características extraídas em cada abordagem, nas bases de treinamento e teste. O gráfico foi plotado utilizando TSN para redução de dimensionalidade.

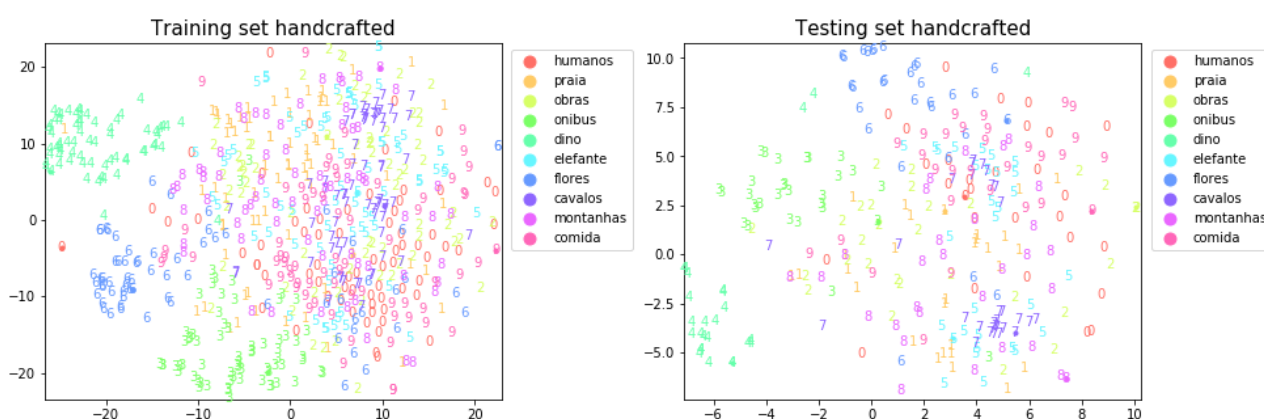


Figura 1 - Base de treinamento e validação handcrafted.

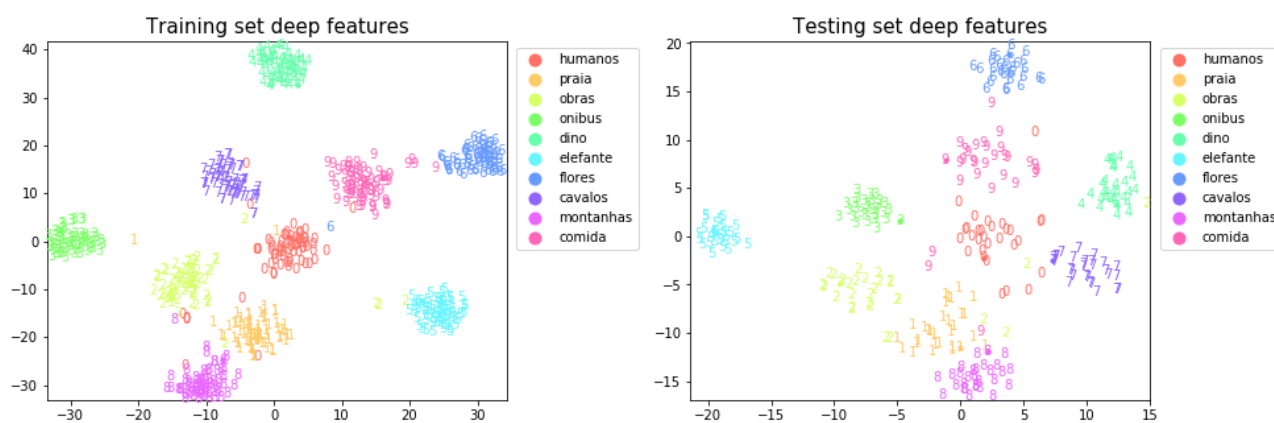


Figura 2 - Base de treinamento e validação deep features.

C. Matriz de confusão dos melhores resultados:

Conforme a acurácia dos classificadores, o melhor resultado obtido foi através de Logistic Regression na base de características extraídas via deep learning.

Confusion Matrix LogisticRegression:

```
[ [28 1 0 0 0 0 0 0 0 1]
  [ 0 30 0 0 0 0 0 0 0 0]
  [ 0 1 28 0 1 0 0 0 0 0]
  [ 0 0 0 30 0 0 0 0 0 0]
  [ 0 0 0 0 30 0 0 0 0 0]
  [ 0 0 0 0 0 30 0 0 0 0]
  [ 0 0 0 0 0 0 30 0 0 0]
  [ 0 0 0 0 0 0 0 30 0 0]
  [ 0 0 0 0 0 0 0 0 30 0]
  [ 0 0 0 0 0 0 0 0 0 30]]
```

Figura 3 - Matriz confusão classificador Logistic Regression.

SVM também apresentou um bom resultado na base deep features.

Confusion Matrix SVM:

```
[[28  1  0  0  0  0  0  0  0  1]
 [ 0 30  0  0  0  0  0  0  0  0]
 [ 1  1 28  0  0  0  0  0  0  0]
 [ 0  0  0 30  0  0  0  0  0  0]
 [ 0  0  0  0 30  0  0  0  0  0]
 [ 0  0  0  0  0 30  0  0  0  0]
 [ 0  0  0  0  0  0 30  0  0  0]
 [ 0  0  0  0  0  0  0 30  0  0]
 [ 0  0  0  0  0  0  0  0 30  0]
 [ 1  0  0  0  0  0  0  0  0 29]]
```

Figura 4 - Matriz confusão classificador SVM.

d. Análise de erros:

Erros apresentados no classificador LogisticRegression – Base deep features:

Label: 2 obras / Prediction: 1 praia proba 0.3929105845786363

/content/drive/My Drive/Base/obras/297.jpg

Label: 0 humanos / Prediction: 9 comida proba 0.9296879822790137

/content/drive/My Drive/Base/humanos/75.jpg

Label: 2 obras / Prediction: 4 dino proba 0.575441287886532

/content/drive/My Drive/Base/obras/299.jpg

Label: 0 humanos / Prediction: 1 praia proba 0.7160476443821734

/content/drive/My Drive/Base/humanos/72.jpg

Erros apresentados no classificador SVM – Base deep features:

Label: 2 obras / Prediction: 1 praia proba 0.4870863374977947

/content/drive/My Drive/Base/obras/297.jpg

Label: 0 humanos / Prediction: 9 comida proba 0.5614931479181082



Pró-Reitoria de Pesquisa, Pós-Graduação e Inovação

Diretoria de Educação Continuada

Pós-Graduação Lato Sensu

Curso de Especialização em Inteligência Artificial Aplicada

Módulo Aprendizagem de Máquina

/content/drive/My Drive/Base/humanos/75.jpg

Label: 2 obras / Prediction: 0 humanos proba 0.4461696041099221

/content/drive/My Drive/Base/obras/299.jpg

Label: 9 comida / Prediction: 0 humanos proba 0.30589354088648774

/content/drive/My Drive/Base/comida/903.jpg

Label: 0 humanos / Prediction: 1 praia proba 0.3916637647673335

/content/drive/My Drive/Base/humanos/72.jpg

5. Conclusão

Conforme análise dos dados obtidos, observou-se que as características extraídas manualmente não estão bem separadas, o que dificulta muito a criação da linha de decisão por parte dos classificadores, justificando o pior desempenho dos classificadores a partir desta abordagem.

Através da análise dos erros apresentados, é possível observar que a probabilidade apresentada (proba) na maioria deles é baixa, fato este que através da análise do parâmetro, possibilita que seja empregado outro classificador que apresente resultados melhores para o caso específico. Além disso, através de uma breve análise da base de imagens, é possível encontrar imagens com o rótulo errôneo, como por exemplo humanos/72.jpg, cuja imagem não apresenta nenhuma figura humana.