

Processamento Digital de Imagens  
Trabalho prático 7  
Contornos, Momentos de Hu, Zernike e Classificações

Josué Nunes - 3465  
Mateus Coelho - 3488  
Ricardo Spínola - 3471

17 de novembro de 2022

## Conteúdo

<b>1</b>	<b>Introdução</b>	<b>2</b>
<b>2</b>	<b>Conceitos</b>	<b>2</b>
2.1	Momentos invariantes de uma imagem . . . . .	2
2.1.1	Momento de Hu . . . . .	2
2.1.2	Momentos de Zernike . . . . .	2
2.2	Classificadores - SVM . . . . .	2
2.2.1	Kernel . . . . .	2
2.3	Análise de componentes principais - PCA . . . . .	3
2.4	Contornos . . . . .	3
<b>3</b>	<b>Desenvolvimento</b>	<b>3</b>
3.1	<i>Dataset</i> de linguagem de sinais . . . . .	3
3.2	<i>Dataset</i> de logotipos . . . . .	4
<b>4</b>	<b>Resultados</b>	<b>4</b>
4.1	Gestos . . . . .	5
4.1.1	Momento Hu . . . . .	5
4.1.2	Momento de Zernike . . . . .	5
4.2	Logo . . . . .	6
<b>5</b>	<b>Conclusão</b>	<b>6</b>

# 1 Introdução

A classificação de imagens é um processo que consiste na extração de informação de uma imagem afim de reconhecer padrões e objetos presentes na mesma. No processo de extração de informação de uma imagem, podem ser utilizadas as técnicas de processamento que dizem respeito a traçar o contorno de objetos, destacá-los e extrair padrões do conjunto de imagens, facilitando o treinamento do modelo de classificação.

O trabalho proposto terá como objetivo implementar um classificador, e usando técnicas conhecidas como momentos invariantes de uma imagem, obter um modelo classificatório de letras em linguagem de sinais e de logotipos de marcas e empresas.

## 2 Conceitos

### 2.1 Momentos invariantes de uma imagem

Em processamento de imagens, momentos são utilizados para caracterizar formas de um objeto. Estes momentos capturam dados como área, centroide e orientação da imagem, de forma que, podem ser utilizados para identificar um objeto, mesmo que tenha sofrido alteração no tamanho, ou rotação.

Em linhas gerais, momentos capturam as características significativas de uma imagem.

#### 2.1.1 Momento de Hu

Momento de Hu denota um conjunto de equações em que os momentos são invariantes a escala, rotação e translação.

$$\begin{aligned}I_1 &= \eta_{20} + \eta_{02} \\I_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\I_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \\I_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \\I_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\I_6 &= (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \\I_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] - (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]\end{aligned}$$

Figura 1: Momentos de Hu

#### 2.1.2 Momentos de Zernike

Utilizando de polígonos ortogonais, os momentos de Zernike têm baixa sensibilidade a adição de ruído, e também são invariantes a escala e rotação. Capaz de caracterizar e quantificar a forma de um objeto, é um dos descritores de imagem mais poderoso, e que adiciona pouco custo computacional.

## 2.2 Classificadores - SVM

Máquinas de vetores de suporte (SVM), utilizadas para classificação e afim, detêm conjunto de técnicas de aprendizado supervisionado.

#### 2.2.1 Kernel

A transformação dos dados de entradas para o hiperplano, juntamente com as modificações necessária, são feitas com álgebra linear. O papel do kernel então é aplicar as transformações, que podem ser:

- linear
- polinomial
- função de base radial (rbf)
- sigmoid

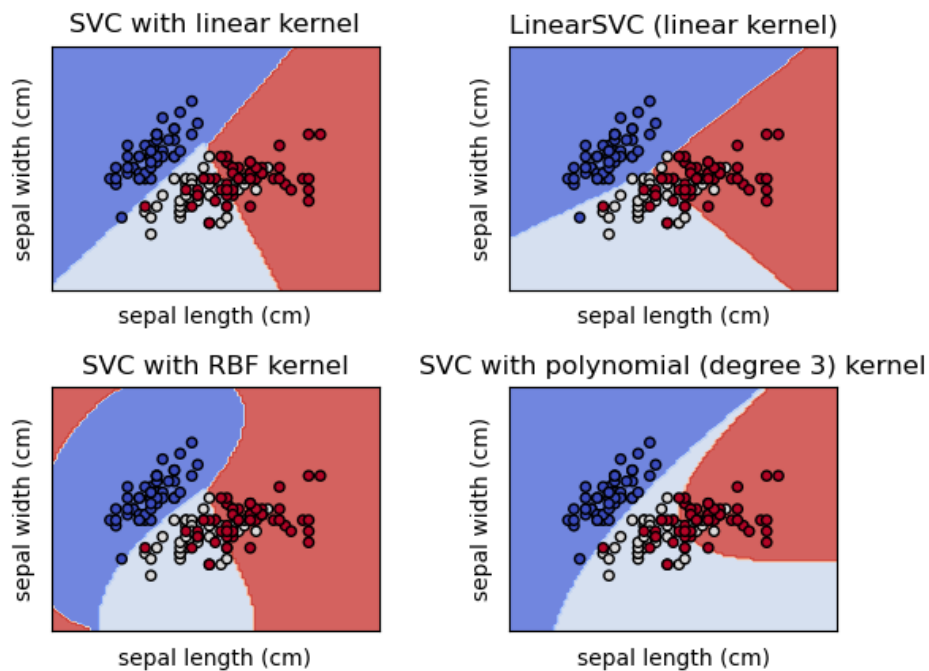


Figura 2: Kernels

### 2.3 Análise de componentes principais - PCA

Método de redução de dimensionalidade frequentemente usado para reduzir a dimensionalidade de um grande conjunto de dados.

### 2.4 Contornos

Contornos são, basicamente a tentativa de encontrar uma curva que junta os pontos contínuos da superfície de um objeto, ou seja, pontos que possuem mesma cor ou intensidade nessa superfície.

## 3 Desenvolvimento

Nesta seção descrevemos o processo de desenvolvimento e todas as tomadas de decisões que estabelecemos para obtermos os resultados esperados. Dividimos esta seção em dois tópicos importantes, o desenvolvimento para tratarmos o conjunto de dados de linguagem de sinais e para tratarmos o conjunto de dados de logotipos.

### 3.1 *Dataset* de linguagem de sinais

Neste conjunto de dados, trabalhamos com as letras de A até N, resultando em uma quantidade de em média 2000 imagens por letra. Em seguida, estabelecemos o método de processar as imagens de forma que os momentos de Hu e Zernike pudessem ser gerados da melhor maneira possível para o classificador SVM. Sendo assim, nossa melhor abordagem baseou-se em utilizar o filtro gaussiano para reduzir os ruídos das imagens, com um *kernel* de 5x5 para extrair o momento de Zernike sem PCA, e um *kernel* 7x7 para extrair os momentos de Hu e Zernike com PCA.



Figura 3: Imagens do dataset de gestos processadas

Posteriormente, aplicamos a técnica de binarização de imagem utilizando o método *adaptive threshold* da biblioteca OpenCV. Decidimos utilizar este método, pois foi o que obteve melhores resultados durante nossos testes. Além

disso, invertemos a imagem de maneira que o fundo permanecesse preto e o objeto branco para os momentos de Hu e Zernike com PCA, porém não realizamos a inversão no momento de Zernike sem PCA para melhores resultados.

Com o pré-processamento das imagens realizado, conseguimos calcular os momentos de Hu e Zernike sem limitações, utilizando as próprias bibliotecas disponíveis no Python. No caso do momento de Hu, após o cálculo do momento, aplicamos a transformação logarítmica para normalizar os valores em um intervalo que fosse melhor para o classificador trabalhar:

$$H_i = -\text{sign}(h_i) * \log(|h_i|)$$

. Já para o momento de Zernike não foi necessária nenhuma transformação, exceto que na utilização do PCA os momentos são alterados com a utilização desta técnica, conforme visto anteriormente.

Partindo dos momentos gerados, o nosso conjunto de dados pode ser exportado para um arquivo .csv, contendo os momentos para cada imagem e a identificação de qual imagem um determinado momento é, de maneira que o classificador pudesse ser treinado com estes dados. Logo, o SVM fez uso do *kernel* rbf, pois foi o que se sobressaiu em nossos testes. Adicionalmente, utilizamos os seguintes hiperparâmetros para cada momento. A escolha dos hiperparâmetros ideais se deu pela execução de várias rodadas de teste até obtermos as melhores precisões sem causar *overfitting*:

1. Momento de Hu: Regularização = 100 e *gamma* = 1;
2. Momento de Zernike sem PCA: Regularização = 100 e *gamma* = 50;
3. Momento de Zernike sem PCA: Regularização = 100 e *gamma* = 0.1;

### 3.2 Dataset de logotipos

Neste conjunto de dados, foram utilizadas as mesmas estratégias do conjunto anterior. Ou seja, utilizamos apenas o momento de Hu para este caso, sendo que não foi necessária nenhuma alteração no cálculo do momento, tampouco no modelo SVM. Um ponto negativo para este conjunto de dados foi a limitação da quantidade de imagens de logotipos, dificultando o treinamento do modelo e, conseqüentemente, sua precisão. De qualquer maneira, conseguimos obter alguns resultados descritos na seção seguinte.



Figura 4: Imagens do dataset de logotipos processadas

Sendo assim, com o desenvolvimento concluído, foi possível executar várias rodadas de testes com imagens de fora do conjunto de dados para avaliarmos os resultados da implementação.

## 4 Resultados

As imagens abaixo mostram os resultados encontrados depois de treinar o modelo. Para o dataset de logotipos, usamos a seguinte imagem, tanto no momento de Hu, quanto Zernike.



Figura 5: Figura utilizada nos testes de gestos

## 4.1 Gestos

### 4.1.1 Momento Hu

```
Gerando momentos...  
Exportando dados...  
Treinando o modelo...  
Precisao do modelo: 96.247 %  
Predizendo imagem...  
Esta imagem e uma letra C!
```

Figura 6: Momento Hu - Modelo

### 4.1.2 Momento de Zernike

```
Gerando zernike momentos...  
Exportando dados...  
Treinando o modelo...  
Precisao do modelo: 84.395 %  
Predizendo imagem...  
Esta imagem e uma letra C!
```

Figura 7: Momento Zernike - Modelo sem PCA

```
Gerando zernike moments com pca...  
Exportando dados...  
Treinando o modelo...  
Precisao do modelo: 93.317 %  
Predizendo imagem...  
Esta imagem e uma letra C!
```

Figura 8: Momento Zernike - Modelo com PCA

## 4.2 Logo

```
Gerando momentos...  
Exportando dados...  
Treinando o modelo...  
Precisao do modelo: 48.571 %  
Predizendo imagem...  
Esta logo e apple
```

Figura 9: Momento Hu - Modelo

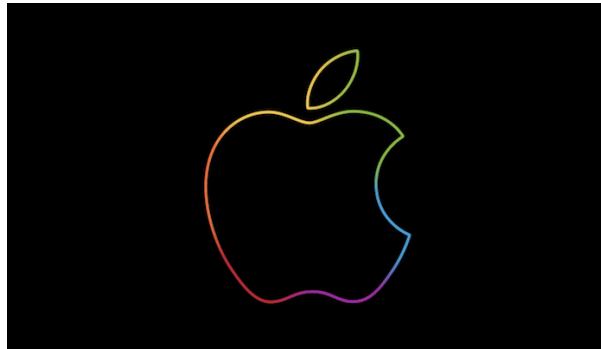


Figura 10: Predição correta para a logo da apple

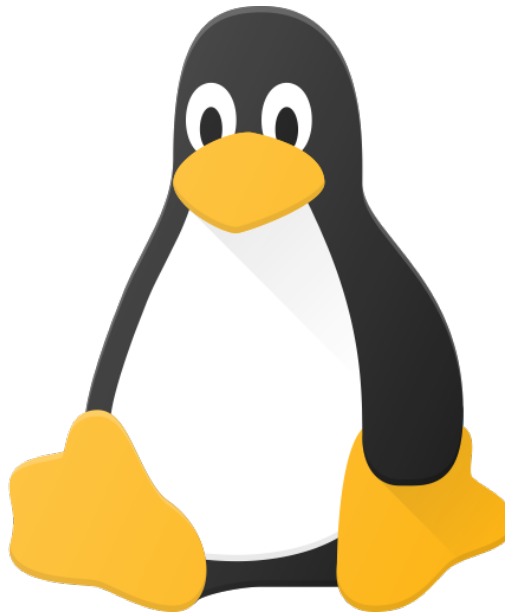


Figura 11: Predição correta para a logo do linux

## 5 Conclusão

Os modelos criados classificam bem em geral, para o dataset de gestos, mas por falta de uma variedade de amostras, torna-se ineficaz a classificação de um gesto fora do contexto que é apresentado, por exemplo, fora de um fundo branco, ou mesmo com mais objetos ao redor.

Já o modelo criado para prever logotipos tem baixa precisão, resultado esse encontrado pela falta de dados para treinamento do modelo.