

Uma framework para a classificação de objetos

Aluno: Felipe Añez de Toledo Blassioli Orientador: Roberto Hirata Junior

Instituto de Matemática e Estatística (IME USP)

Motivação

O reconhecimento de objetos em imagens é um problema antigo e conhecido que continua sendo de muito interesse nos dias de hoje. Prova disso são tecnologias emergentes como carros sem motorista que devem fazer detecção de obstáculos em tempo-real.

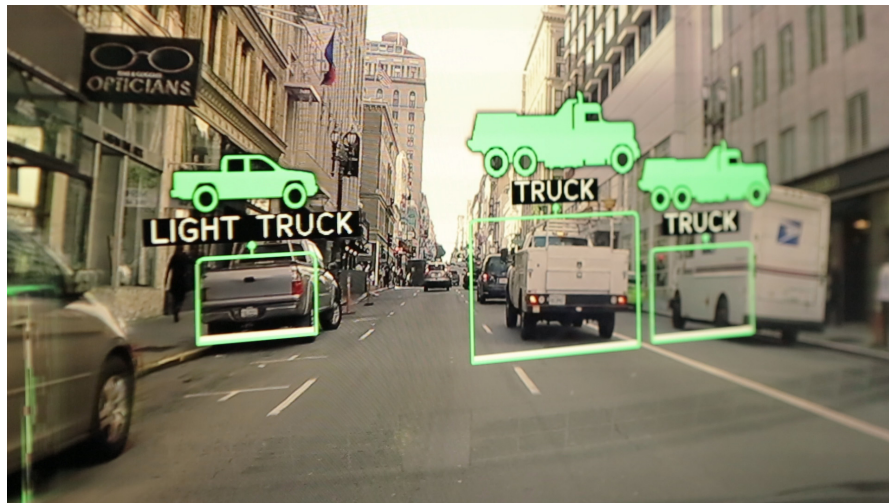


Figura 1: Carros sem motorista [1]

Objetivo

Desenvolver uma *framework* que faça uso de ferramentas de processamento de imagens e aprendizado de máquina disponíveis gratuitamente para construir soluções para o problema de reconhecimento de objetos e que automatize etapas repetitivas desse processo tais como a leitura de datasets e geração de métricas facilitando a comparação entre diferentes abordagens via análise de resultados de experimentos realizados sistematicamente a fim de encontrar a melhor solução para o problema.

A Framework

Foi desenvolvida uma *framework* em linguagem python que abstrai a noção de experimento como o conjunto de três elementos: *Dataset*, *DescriptorExtractor* e *Classifier*. O *Dataset* constitui as imagens de treinamento e teste, o *DescriptorExtractor* é responsável por transformar uma imagem num vetor de números e o *Classifier* é responsável por construir um modelo capaz de fazer previsões. E que disponibiliza por padrão algoritmos conhecidos como: SIFT, SURF, MSER, SVM e outros como o descritor GRABED [2].

Realizando um experimento

A framework fica disponível como pacote Python *senjo*. E criar um experimento é tão simples quanto estender uma das classes de experimento do pacote e definir qual extrator de características, classificador e dataset utilizar.

```
from senjo.experiments import Experiment, BOWExperiment
from senjo.algorithms import GRABED, SVM
```

```
class GRABEDExperiment(Experiment):
    @property
    def classifier(self):
        return SVM(C=1, gamma=0.5)
    @property
    def descriptor_extractor(self):
        return GRABED()

class BOWGRABEDExperiment(BOWExperiment):
    @property
    def descriptor_extractor(self):
        return GRABED()
```

O exemplo acima cria dois experimentos que fazem uso do descritor GRABED e do classificador SVM. A diferença entre eles é que um utiliza a abordagem Saco de Palavras [3] e o outro não.

Visualizador de Resultados

O visualizador de resultados incluso no pacote da *framework* consiste num aplicativo web que permite ao usuário avaliar o desempenho de um experimento classe a classe, bem como comparar o resultado de diferentes experimentos. Para tal fim ficam disponíveis as seguintes informações:

- Matriz e Tabela de Confusão
- Medidas estatísticas comuns como: precisão, acurácia, F1, MCC, entre outras.

ACC	81.4 %	FDR	18.6 %
F1	79.5 %	PPV	81.4 %
FNR	20.8 %	FPR	15.9 %
TNR	84.1 %	NPV	82.9 %
MCC	63.8 %	TPR	79.2 %

Figura 2: Medidas estatísticas

Comparação de Experimentos

Essas medidas também podem ser visualizadas como um gráfico de linhas e comparadas entre experimentos. Sendo útil para encontrar os parâmetros ótimos de um algoritmo de acordo com uma métrica como, por exemplo, precisão. A figura abaixo, mostra a diferença de diversas métricas entre experimentos que variam os parâmetros C e γ do classificador SVM de acordo com uma escala logarítmica.

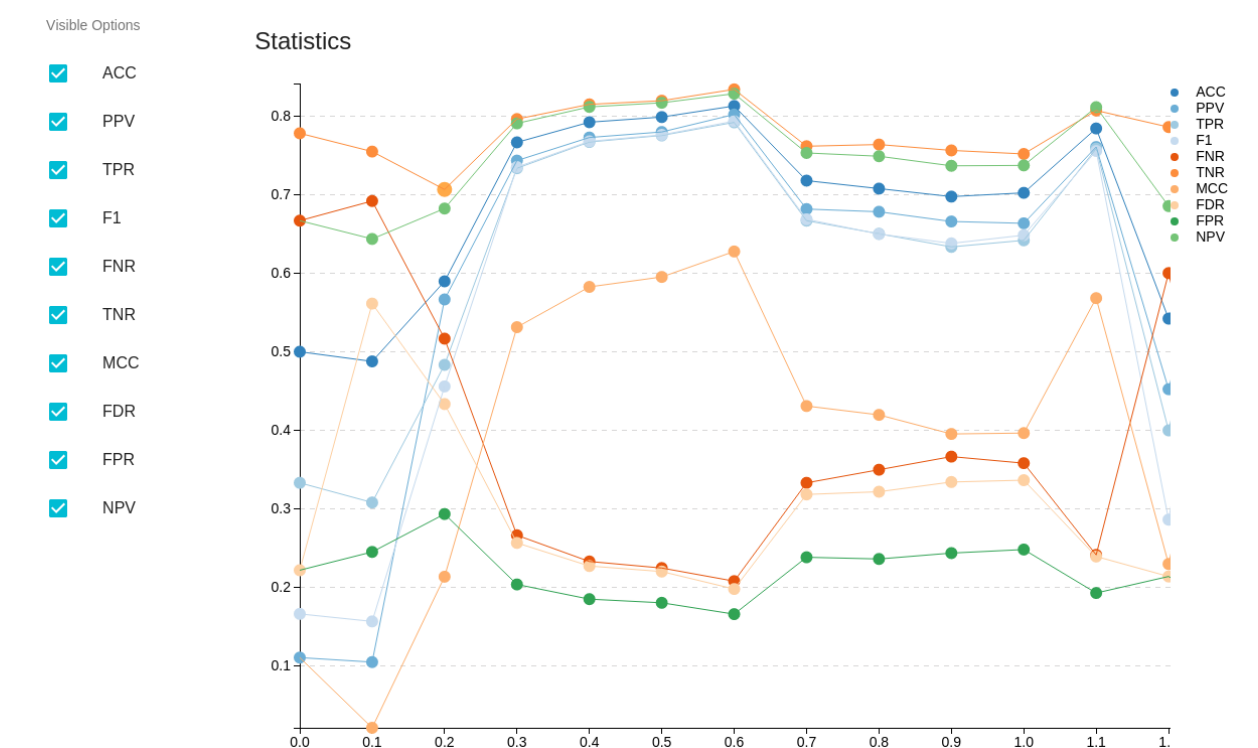


Figura 3: Comparação de experimentos: variando parâmetros do SVM

Trabalhos Futuros

- Adicionar mecanismos de *cross-validation* e suporte a outras bibliotecas populares como VLFeat e SimpleCV à framework.
- Transformar o visualizador de resultados numa aplicação capaz de realizar experimentos, isto é, fazendo uso da framework.

Referências

- [1] Nvidia announces pricing and a launch date for drive px self-driving car platform at gtc 2015, 2015.
- [2] Roberto Hirata Jr Arnaldo Câmara Lara. A granulometry based descriptor for object categorization. *Mathematical Morphology and Its Applications to Signal and Image Processing*, pages 413–424, 2013.
- [3] Andrew Zisserman Josef Sivic. Video google: A text retrieval approach to object matching in videos. *ICCV '03 Proceedings of the Ninth IEEE International Conference on Computer Vision*, page 1470, 2003.