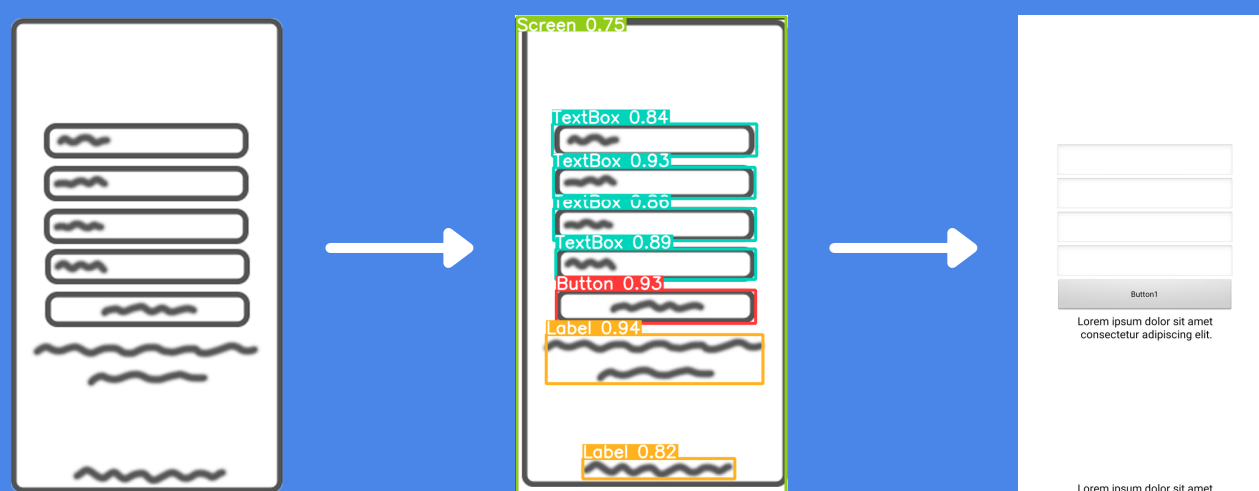


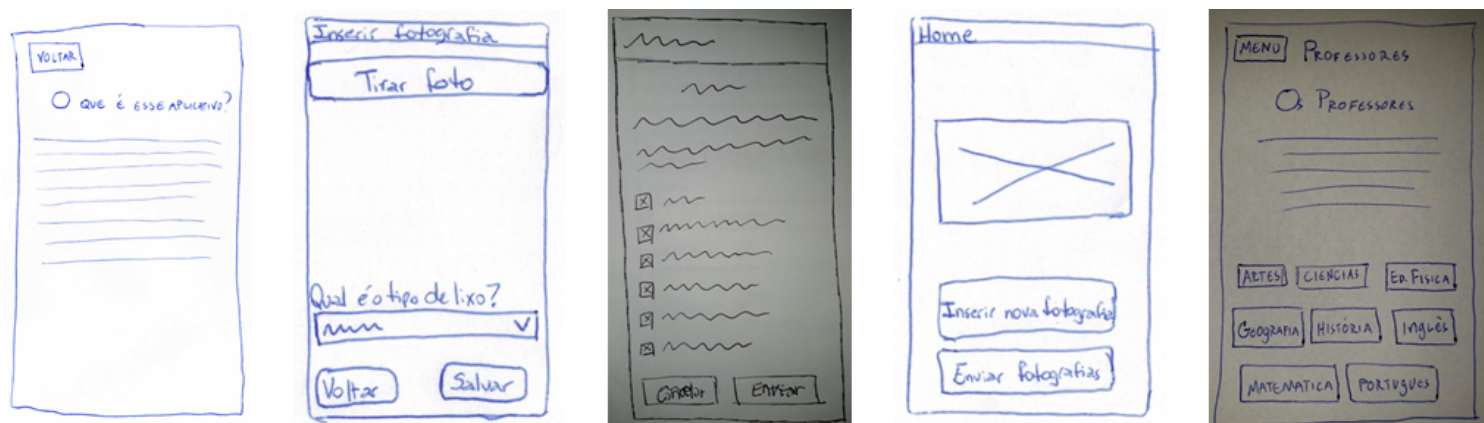
Sketch2aia

Geração automática de wireframes em App Inventor a partir de sketches



O que são Sketches?

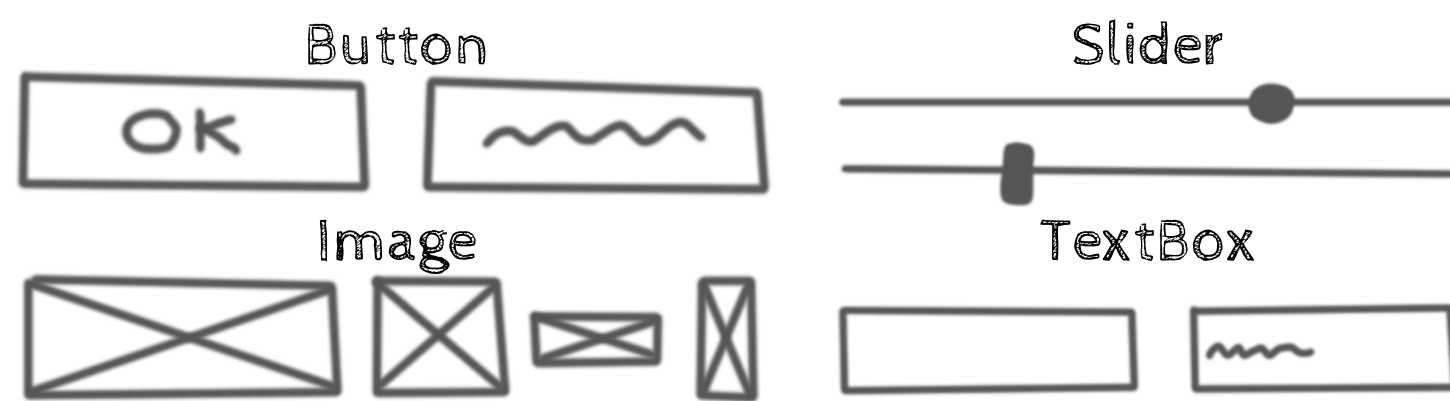
Sketches são rascunhos, geralmente em papel e caneta, muito utilizados no processo de design de interfaces gráficas como uma forma de transmitir ideias e testar conceitos:



Uma vez que um sketch é aprovado, cabe a um programador realizar manualmente sua implementação como wireframe em computador, para que suas funcionalidades possam ser testadas. Posteriormente, elementos de design visual são adicionados, resultando na interface refinada de um produto.

Desafio Proposto

Para automatizar a conversão de sketches de interfaces de usuários em wireframes de App Inventor, temos o desafio de detectar automaticamente a posição e o tipo dos componentes de interface, por exemplo:

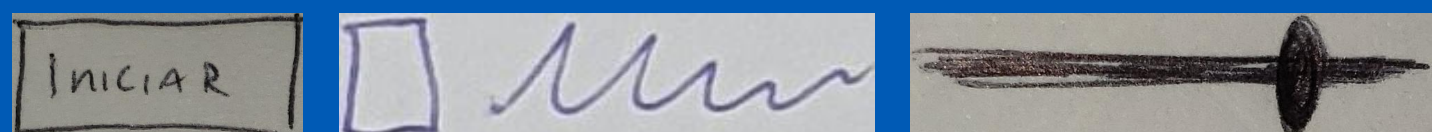


Com os componentes identificados e suas posições, é possível gerar o wireframe em App Inventor automaticamente, sem a necessidade da conversão manual por um desenvolvedor.

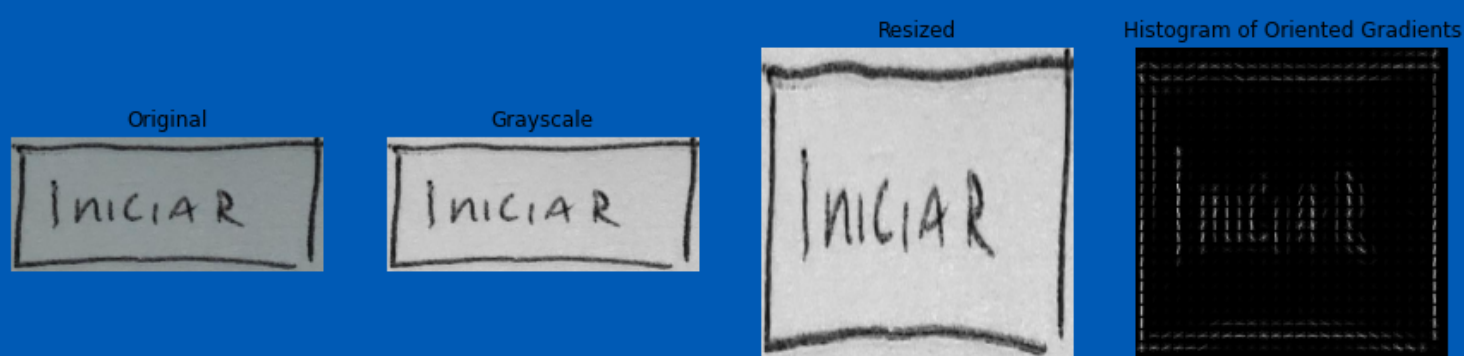
Visão Computacional Clássica - HoG

Para solucionar o problema com técnicas de Visão Computacional Clássicas, foi utilizado o HoG (*Histogram of Gradients*) para a extração de características, seguido de um classificador KNN, para determinação da classe de componente.

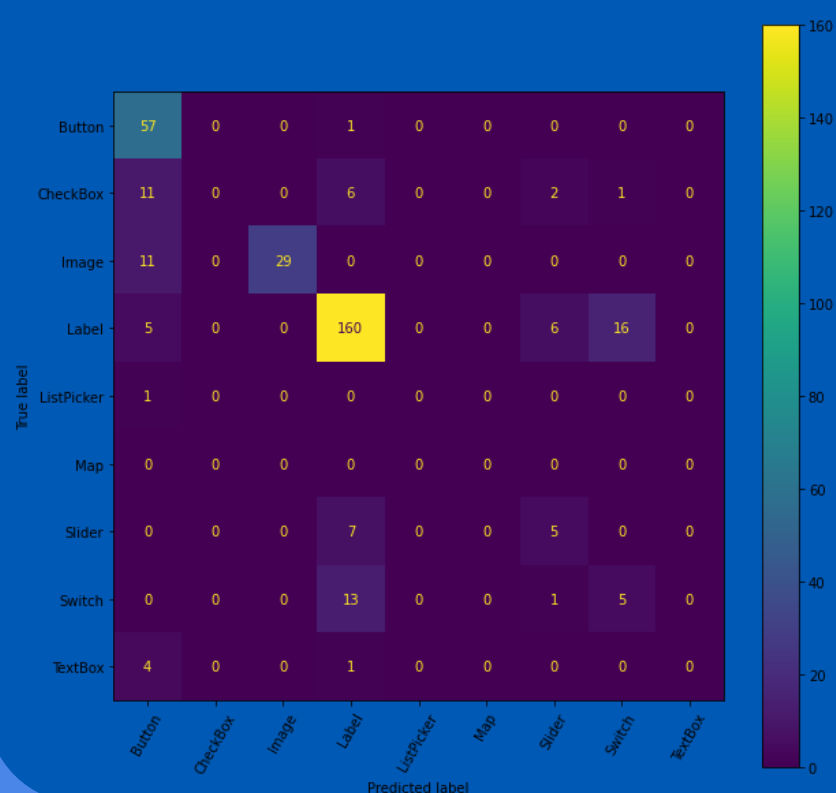
O primeiro passo para isso foi a criação de um conjunto de dados de recortes de sketches de componentes de interface individualmente:



Todas as 3183 imagens do conjunto de dados foram então convertidas para grayscale e redimensionadas para um tamanho padrão. Cada imagem teve seu HoG calculado:



Este conjunto de HoGs foi então utilizado para o treinamento de um *KNeighborsClassifier*, que pode então ser utilizado para a determinação do tipo de um componente, a partir do HoG calculado. O classifier obteve a seguinte performance:

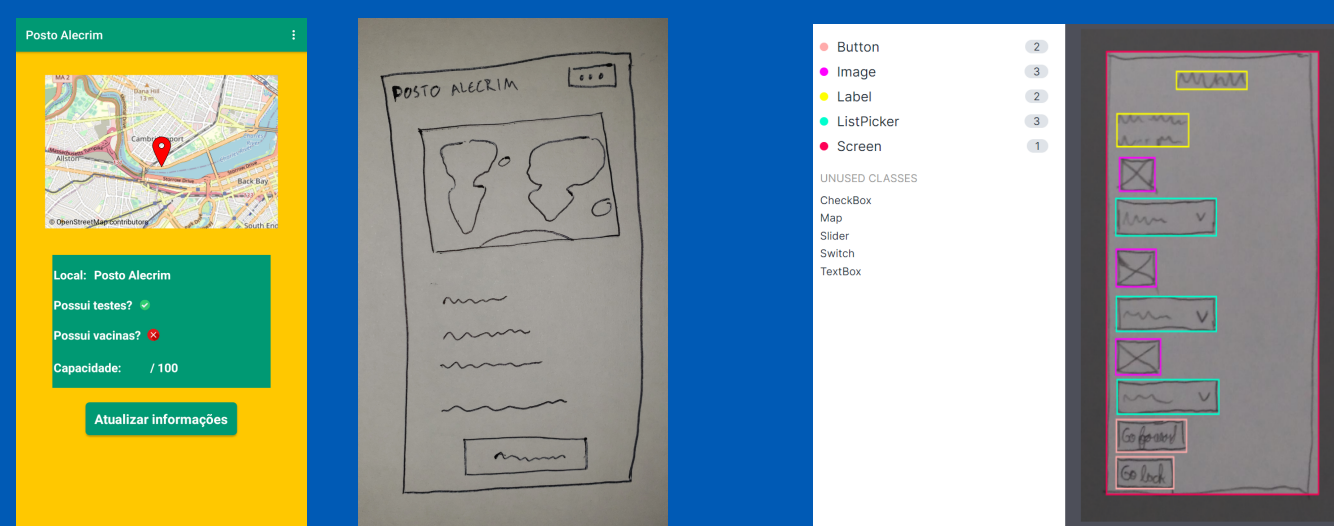


O classificador se saiu bem na classificação de botões, imagens e labels, mas deixou muito a desejar na classificação das demais classes de componentes de interface de usuário do conjunto de dados.

Deep Learning - YOLOv5

Para a abordagem utilizando Deep Learning, se optou pela utilização do YoloV5 devido aos bons resultados obtidos anteriormente com YoloV3 e YoloV4, sua boa performance em geral e o baixo impacto de seus principais pontos fracos no problema em questão.

Foi preparado um conjunto de dados de sketches de interfaces de usuário, feitos com base em screenshot de aplicativos App Inventor, rotulados manualmente com a posição e tipo de seus componentes de interface:



Foi realizado o treinamento de um modelo YOLOv5m6 por aproximadamente 600 épocas, obtendo mAP@0.5 de 0.97 entre todos os componentes, com o pior resultado, switch, obtendo mAP@0.5 de 0.94. A matriz de confusão obtida, bem como um exemplo de detecção podem ser vistos abaixo:

