

Victória Martins

Protocolo de processamento e interpretação de imagens de psicotestes

OBJETIVO

O Detran (Departamento de Trânsito) é o órgão responsável por habilitar condutores. Em uma das etapas de avaliação, psicotestes são aplicados a fim de avaliar a habilidade psicológica dos candidatos. Dentre estes testes, é possível destacar um popularmente conhecido como “teste dos tracinhos”, em que o candidato deve desenhar traços verticais por um período de tempo e, quando o psicólogo sinaliza um som combinado entre as partes, um traço horizontal deve ser desenhado. Assim, este trabalho tem como objetivo implementar um protocolo de processamento e interpretação de imagens que contabiliza quantos traços foram desenhados utilizando a linguagem de programação Python.

FERRAMENTAS UTILIZADAS

Testes foram simulados utilizando papel sulfite A4 e caneta azul. Para a obtenção das imagens, foi utilizada a câmera do smartphone Xiaomi Redmi Note 9, que possui 48 megapixels de resolução.

Em sequência, as imagens foram cortadas para manter apenas a informação do papel por meio do editor de fotos do Windows 10. As imagens resultantes foram importadas e processadas em um arquivo codificado em Python desenvolvido na IDE PyCharm com auxílio das bibliotecas NumPy, OpenCV e Imutils.

DESAFIOS

O principal desafio durante o desenvolvimento derivou das imprecisões que a caneta deixou no papel. Isso acontece porque em alguns pontos dos traços (principalmente onde ele se inicia), a tinta fica mais clara e um único traço seria contabilizado como dois, comprometendo o resultado final. Ademais, a luz incidente sobre o papel se configurou como desafio, uma vez que o flash da câmera produziu sombras. A solução para este cenário foi utilizar luz ambiente, sem flash e sem sombras sobre o papel.

PROTOCOLO

Aquisição

Ao reproduzir o experimento, o dispositivo de captura deve ter qualidade superior a 12 megapixels para que a resolução não afete o resultado final. É necessário focalizar a imagem e cortar as bordas caso algum outro objeto seja capturado.

Nesta etapa, é importante cuidar para que a folha não esteja rasurada nem amassada e que, no momento da captura, exista outra folha branca por baixo do teste para que a transparência não prejudique o processamento.

Pré-processamento e Segmentação

A etapa de pré-processamento iniciou na importação das imagens ao convertê-las para tons de cinza. Em sequência, foram limiarizadas pelo algoritmo de Otsu: tons de cinza abaixo de 100 assumiram valor 0 (cor preta) e os demais tons assumiram valor 255 (cor branca).

A técnica de erosão foi aplicada sobre as limiarizações com o objetivo de expandir pixels pretos. Esta etapa é fundamental para a contagem, pois corrige as falhas causadas pela tinta da caneta (supracitada na seção “Desafios”). Assim, o kernel se configurou como um vetor de cinco linhas, uma coluna e com valor “1” em todas as posições. Veja a diferença nas imagens 1 e 2:

Imagem 1: Imagem limiarizada de psicoteste

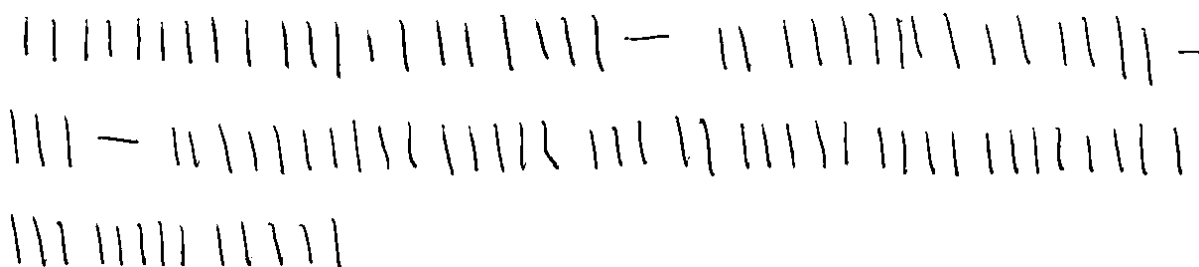
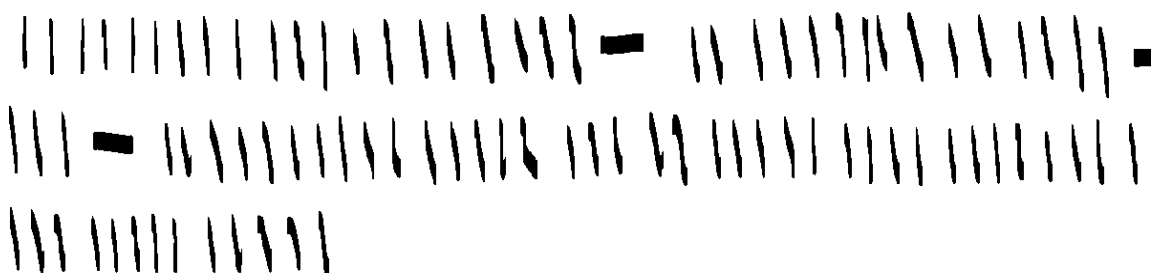


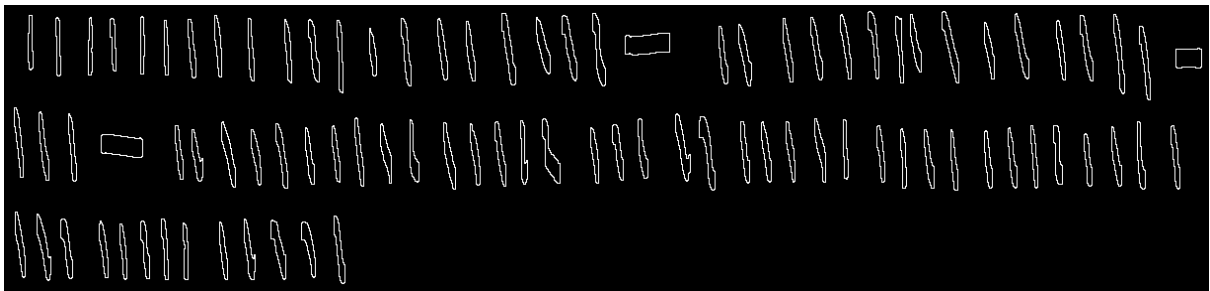
Imagem 2: Erosão aplicada sobre imagem limiarizada



É possível perceber que, por conta do kernel 5x1, os traços ficaram mais altos e, em menor escala, mais grossos.

A próxima etapa aplicou mais transformações morfológicas às imagens limiarizadas. Considerando que o objetivo destas transformações era produzir bordas, o kernel foi modificado para uma matriz 2x2 com valor “1” em todos os índices e, para de fato reconhecer as bordas, foi utilizado o método `morphologyEx()` da biblioteca OpenCV. O resultado do processamento é apresentado na imagem 3.

Imagem 3: Extração de bordas



Interpretação

A interpretação foi feita utilizando o método `findContours()` e a função `grab_contours()` das bibliotecas OpenCV e `Imutils` respectivamente. A aplicação de `findContours` com os parâmetros `RETR_EXTERNAL` e `CHAIN_APPROX_SIMPLE` encontra apenas o contorno externo de cada objeto verificando sua continuidade na vertical, horizontal e diagonal. Em sequência, `grab_contours` extrai os contornos encontrados retornando um objeto cujo tamanho representa a quantidade de traços.

EXPERIMENTO PARA VALIDAÇÃO DO MODELO

O protocolo foi reproduzido em cinco imagens com o objetivo de verificar sua corretude. Os testes se diferenciavam quanto ao número de traços, número de linhas e distância entre os desenhos. Os traços foram contados manualmente antes de serem executados pelo algoritmo, que acertou a quantidade nos cinco testes.

O código e as imagens estão disponíveis em <https://github.com/victoriamartins/contagem-objetos>.