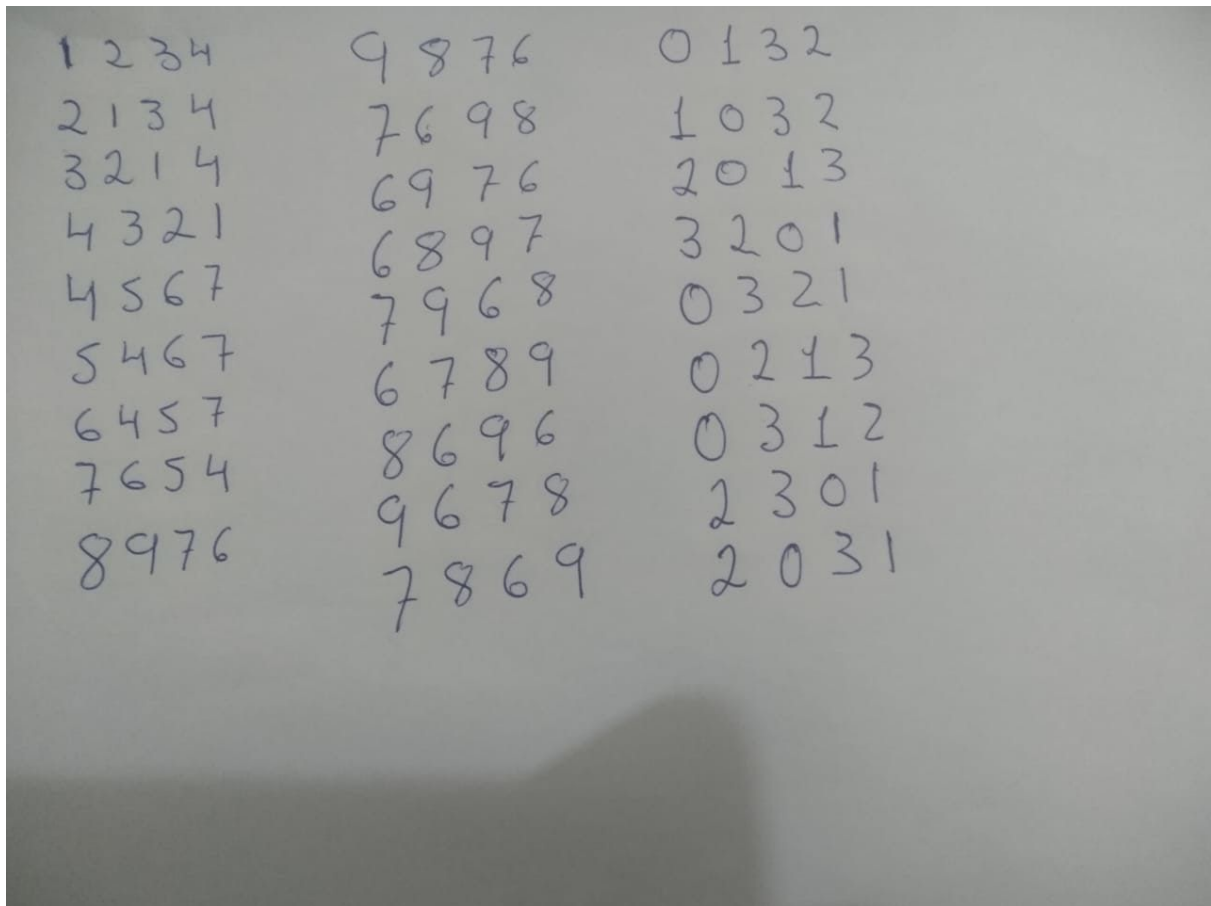


## Processo de desenvolvimento da aplicação:

### Relembrando a ideia:

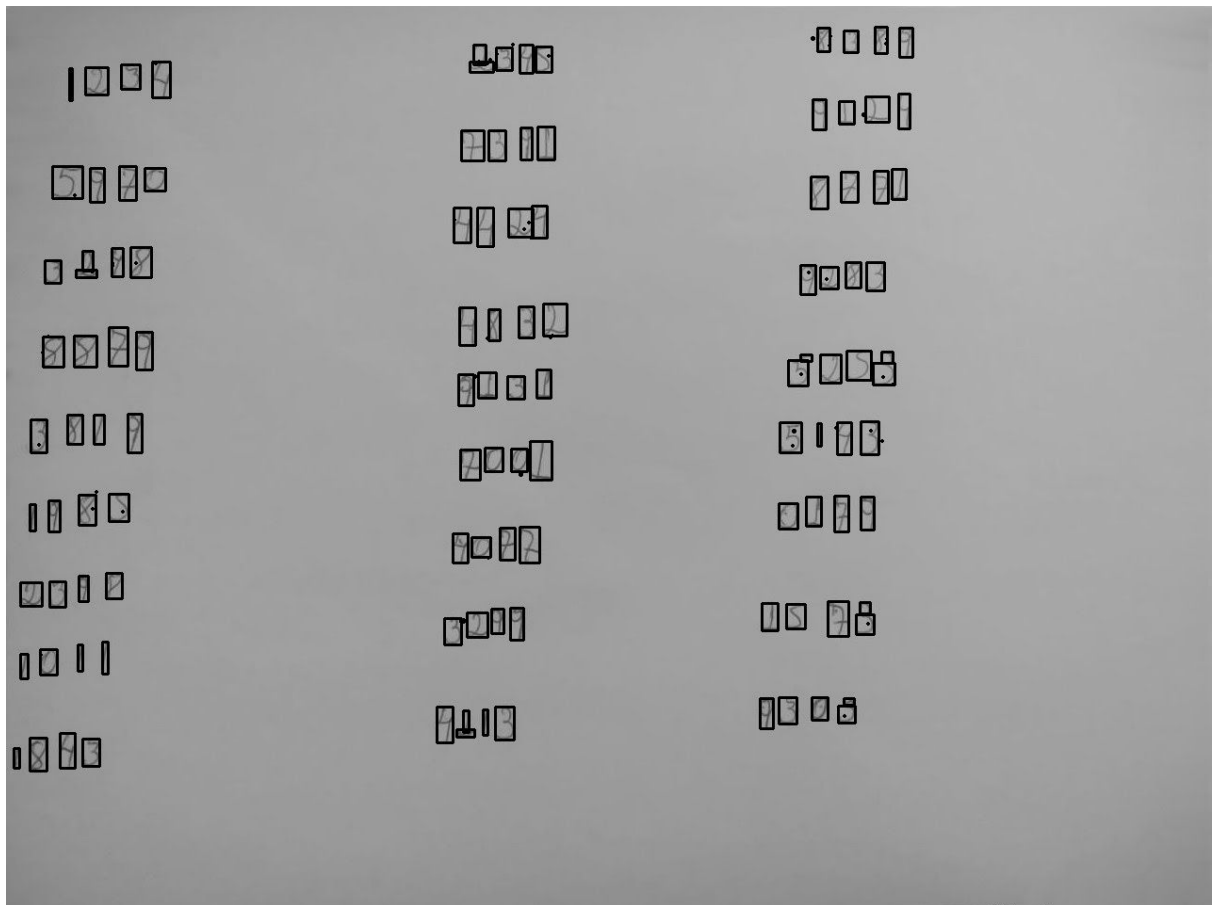
A aplicação recebe uma imagem como entrada, para então exibir como saída o número que possui na folha. Ex: 1234, 2134, 7869...2031



Então, como listar quais números estão nessa folha?

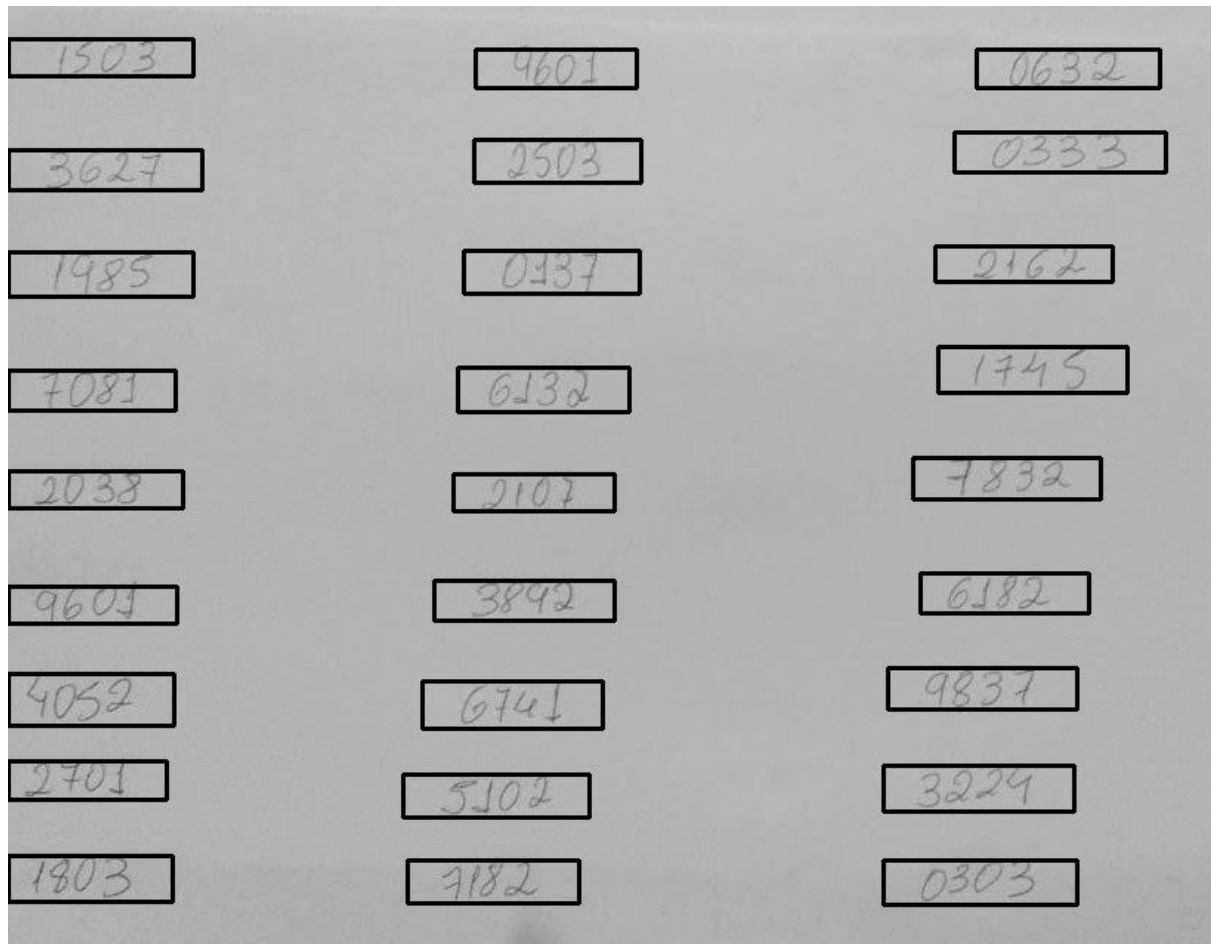
### **Etapa de segmentação:**

Nossa ideia inicial era segmentar todos os números separadamente como mostra a foto abaixo:



Contudo trabalhando dessa maneira se tornava mais complexo a ordenação de cada contorno.

Então utilizamos outra estratégia que consistia em primeiro encontrar os blocos de números agrupados. Para só então encontrar a segmentação separada de cada número.



Dividir o problema diminuía a complexidade de como organizar os números e quais eram os grupos de números que estavam escritos na folha.

Apesar da simplificação do problema, ainda existia uma dificuldade por ser números escritos à mão, como mostrado na figura abaixo:



A primeira estratégia que tentamos usar para segmentação dos números foi calculando a projeção horizontal dos pixels após a imagem ser binarizada e depois identificar os vales que supostamente poderiam separar os números, contudo imagens como essa trazem dificuldade pela falta do “vale” entre os números.

Contudo utilizamos a função do opencv que permite a identificação de contornos utilizando o algoritmo de Suzuki (Suzuki, S. and Abe, K., *Topological Structural Analysis of Digitized Binary Images by Border Following*. CVGIP 30 1, pp 32-46 (1985))

Então após a aplicação dos contornos conseguimos resultados em imagens semelhantes a essas:



Em um total de 1095 números segmentados, 1069 estavam com qualidades boas. Apesar de não sermos especialistas em controle de qualidade de imagens numéricas e por não conhecermos nenhum método de análise de qualidade numérica, que tenha o mesmo objetivo da NFIQ para digitais, essa estimativa foi feita analisando cada imagem separadamente.

Enquanto 26 números seriam perdidos, pois a segmentação não estava funcionando perfeitamente bem:



Bom, alguns desses exemplos são a captura de rasuras(tinha uma rasura no dataset capturado), outras eram por ter 2 números juntos, o contorno capturava ambos, e outros problemas estava em “pedaços” do número faltando.(*Todas as imagens podem ser vistas na pasta do projeto*)

Uma maneira para melhoria dessa segmentação, seria a aplicação de múltiplos métodos de segmentação, como é sugerido no Artigo de Gattal e Chibani:

[https://www.researchgate.net/publication/282184214\\_Combining\\_Multiple\\_Segmentation\\_Methods\\_for\\_Handwritten\\_Digit\\_Recognition\\_of\\_Algerian\\_Bank\\_Checks](https://www.researchgate.net/publication/282184214_Combining_Multiple_Segmentation_Methods_for_Handwritten_Digit_Recognition_of_Algerian_Bank_Checks)

### **Etapa de previsão:**

Após a etapa de segmentação concluída, nos faltava conseguir classificar o número. Um modelo foi treinado com a base de dígitos da MNIST juntamente com os números que foram gerados no dataset após a segmentação, na tentativa de adaptar o previsor para que ele possa também prever números que estavam em dimensões diferentes de 28x28.

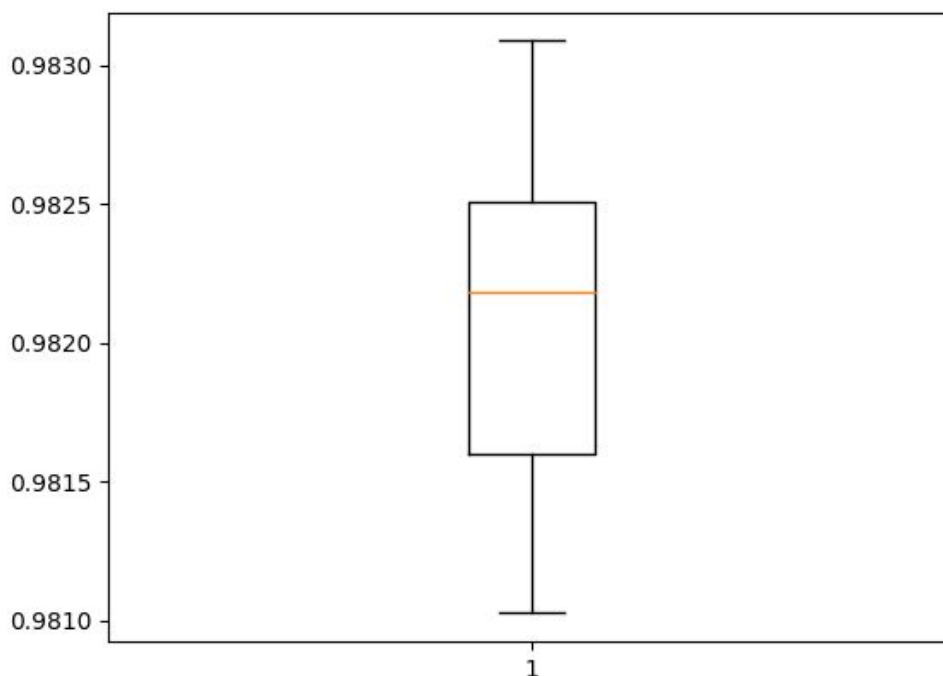
### **Avaliação do modelo:**

O modelo foi avaliado em validação cruzada, onde foi dividido em 5 subconjuntos.

Cada subconjunto obtendo os valores de acurácia de: 98.103 ,98.160, 98.251, 98.218, 98.309

Obtendo uma média de: 98.208 e desvio padrão de: 0.071

Observando o boxplot, podemos ter uma visão melhor do nosso mínimo e máximo e a distribuição dos valores de resultados.



## Avaliação da aplicação:

Apesar do modelo apresentar resultados razoavelmente bons, os resultados na aplicação não foram satisfatórios. Mesmo colocando o dataset que foi extraído para treinamento e teste, a quantidade é insuficiente comparada as imagens do mnist. Testamos observando 3 categorias, sendo elas:

- **Acertos:** A aplicação acertou o bloco de 4 números. Ex: Na lista possui 3202 e a aplicação retornou 3202
- **Erros:** A aplicação errou o bloco de números. Ex: Na lista possui 3202 e a aplicação retornou 3205.
- **Problemas:** Não foram encontrados mais de 4 contornos em um bloco de 4 números

Os resultados na aplicação foram:

- **Taxa de acertos:** 9
- **Taxa de Erros:** 254
- **Taxa de problemas:** 8

## Hipóteses para melhoria:

A base mnist tem suas peculiaridades, como números em imagens 28x28 e números centralizados. Já as imagens capturadas pela segmentação, podem ter tamanhos variados e de alta resolução, ao escalonar as imagens elas podem estar perdendo muita qualidade. Uma hipótese para melhoria então seria adicionar mais imagens a base mnist para o treinamento do modelo, ou então fazer transferência de aprendizado.