

# RealTime Math Solver

Alunos: André Vaz Igarashi

Pablo Rodrigues Sene

# Proposta

- Desenvolver um modelo capaz de detectar e reconhecer equações matemáticas em imagens;
- Reconhecer e resolver a equação em tempo real.

# Etapas do Treinamento

- Gerador de equações utilizando as imagens disponíveis;
- Utilizar as equações geradas para o treinamento da CNN.

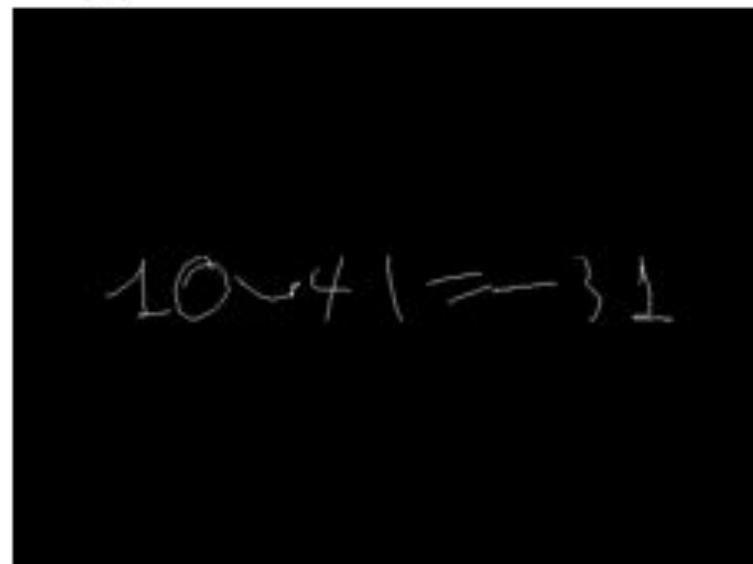
# Pré-processamento de Imagens

- Conversão para tons de cinza usando OpenCV.
- Limiarização adaptativa para criar imagens binárias.
- Inversão da imagem binária para destacar o texto ou símbolos.
- Expansão das dimensões da imagem para compatibilidade com o modelo.

Imagem Binarizada Original

10~41=-31

Imagem Binarizada Invertida



# Preparação dos Dados de Treinamento

- Uso do ImageDataGenerator do Keras para gerar lotes de dados com pré-processamento em tempo real.
- Aplicação da função de binarização como função de pré-processamento.
- Configuração dos parâmetros: diretório de dados, tamanho das imagens, modo de cor, e classe.



# Arquitetura do Modelo de Rede Neural Convolucional (CNN)

- Estrutura da rede: Camadas Convolucionais, Pooling, Flatten e Densas.
- Função de ativação ReLU para introduzir não-linearidade.
- Uso da camada de normalização de entrada (Rescaling).
- Objetivo de cada camada: Extração de características e classificação.

# Treinamento do Modelo de Deep Learning

- Compilação do modelo com o otimizador Adam e função de perda Categorical Crossentropy.
- Métrica de *accuracy* para monitorar a performance do modelo.
- Processo de treinamento usando o gerador de dados de treinamento.
- Duração do treinamento: número de épocas e tamanho do lote.



# Resolução das Equações

- Após a detecção dos números e símbolos presentes uma imagem ou video
- Utiliza o Sympy para resolver a equação encontrada
- Há algum tratamento nas equações encontradas em tempo real

# Processamento do vídeo

- Converte cada frame para *grayscale*
- Aplica detecção de bordas de *Canny*
- Detecta contornos e calcula as caixas delimitadoras utilizando openCV
- Filtra contornos abaixo de um limite mínimo de tamanho e que estejam compartilhando área com contornos maiores

# Demonstração

