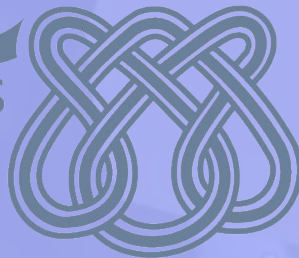


CJ AI

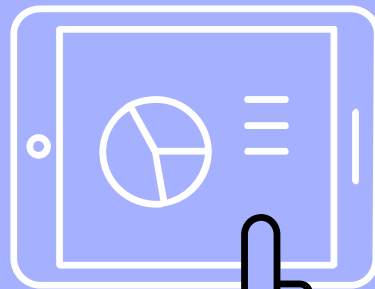
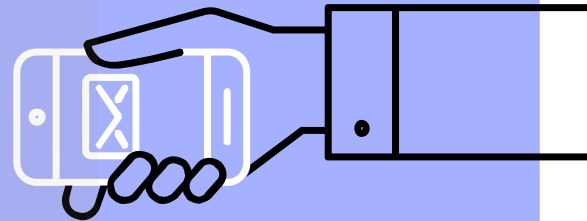
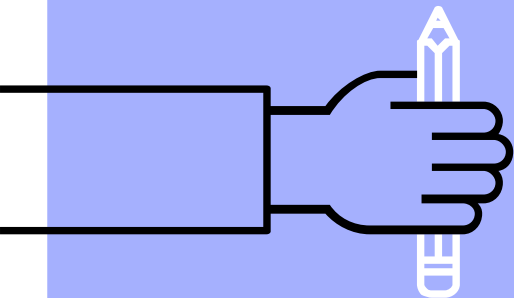
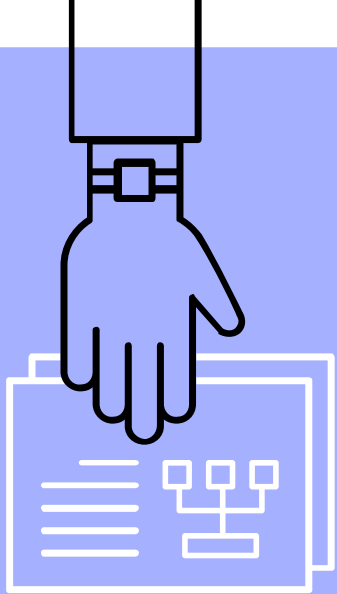
Disciplina: SCC0230 – Inteligência Artificial

Professora: Solange Oliveira Rezende
Estagiário PAE: Vitor Rodrigues Tonon

ICMC USP
SÃO CARLOS

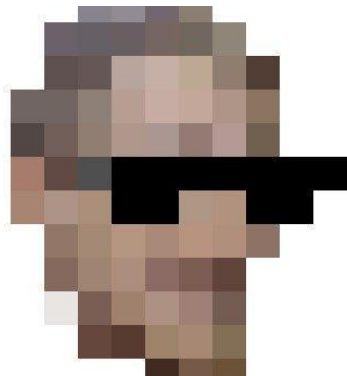


FaceApp

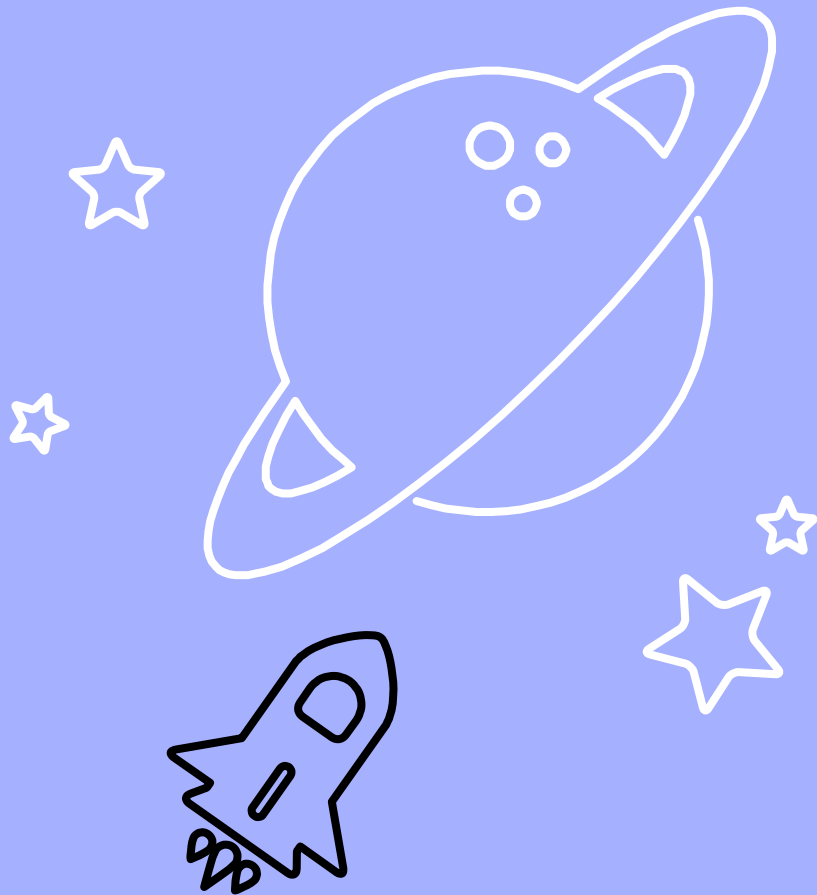


MOTIVAÇÃO

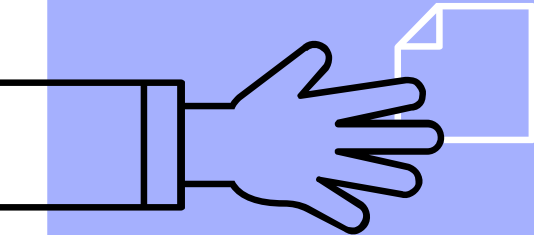
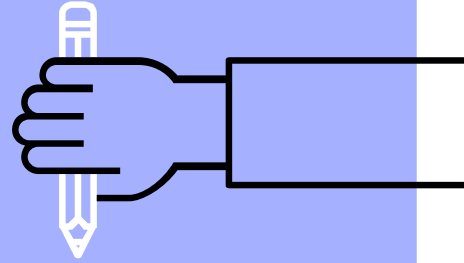
Inspirado na série
*Python plays Grand
Theft Auto V* do
Youtuber Sentdex



GitHub

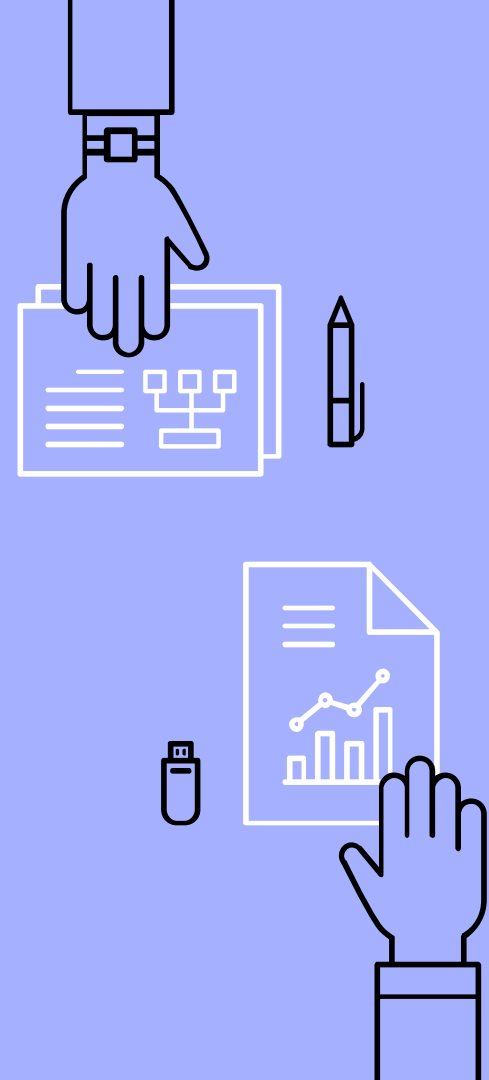


1. IDENTIFICAÇÃO DO PROBLEMA



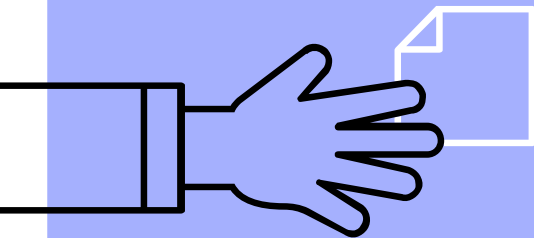
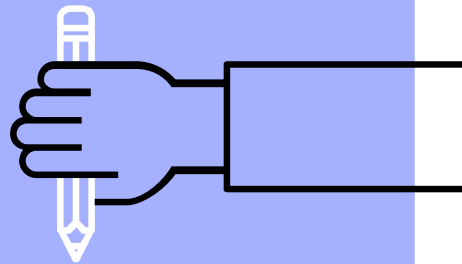
IDENTIFICAÇÃO DO PROBLEMA

- ▶ Simular um motorista no ambiente do jogo
- ▶ Dado a situação do ambiente, o agente deve **escolher** a direção (esquerda, frente, direita)
- ▶ Problema de classificação



2.

PRÉ-PROCESSAMENTO



“

A base de dados utilizada é uma **mistura** de dados coletados pelos integrantes do grupo no GTA San Andreas , consiste de frames do jogo com o rótulo da ação.



EXEMPLO DE DADO

Cada dado consiste de um **frame** e a sua respectiva **ação**.



PRÉ-PROCESSAMENTO

Conversão de RGB para Escala de Cinza

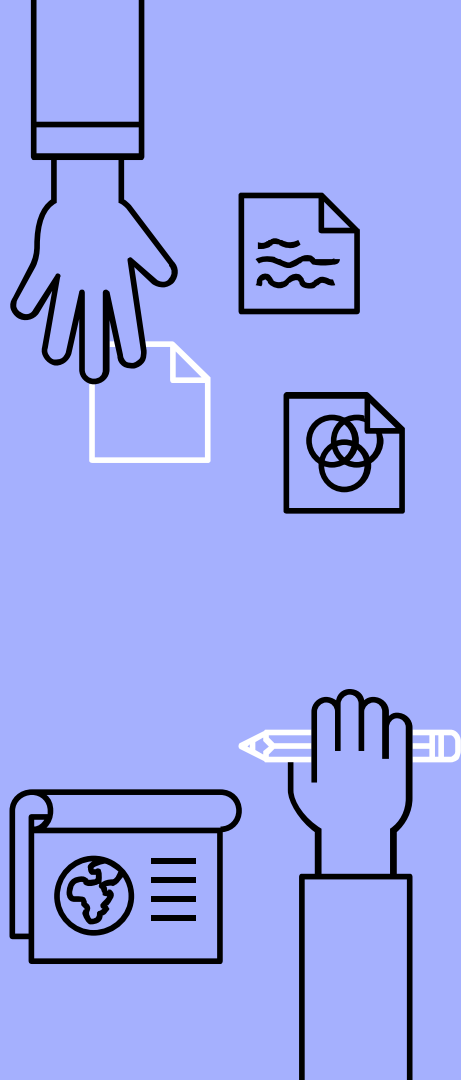
Durante a coleta os dados de imagens são redimensionados e convertidos para cinza para **diminuir a dimensão da imagem** consequentemente diminuindo o tamanho do dado.

Antes: RGB: 800x600x3

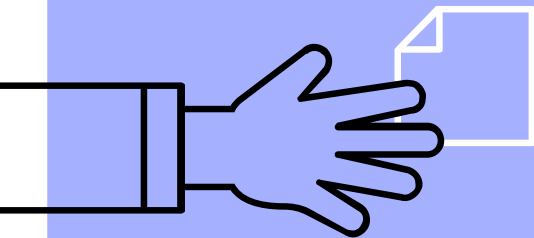
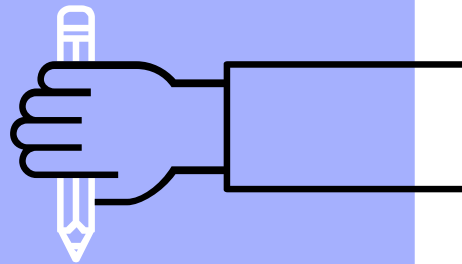
Depois: GRAYSC.: 160x120

Balanceamento dos Dados

Após analisar os dados coletados nota-se que **ir para frente é muito dominante em relação** às outras direções assim o dado utilizado para treinamento possui distribuição igual das direções.



3. EXTRAÇÃO DE PADRÕES



EXTRAÇÃO DE PADRÕES COM DEEP LEARNING

Aprendizado

Redes Neurais Convolucionais

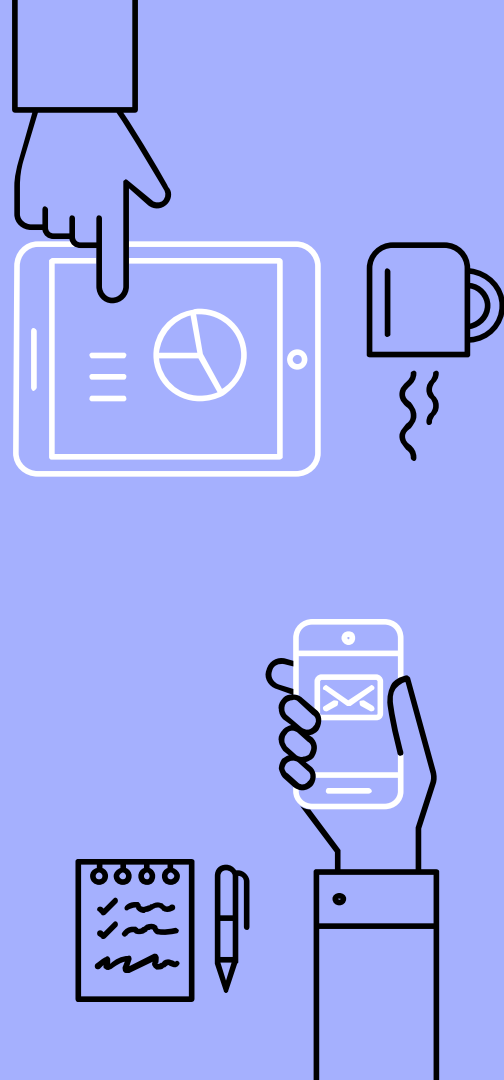
foram utilizadas nos dados pré-processados de modo a aprender e extrair padrões generalizadas que nós humanos temos dificuldades de extrair.

Modelo

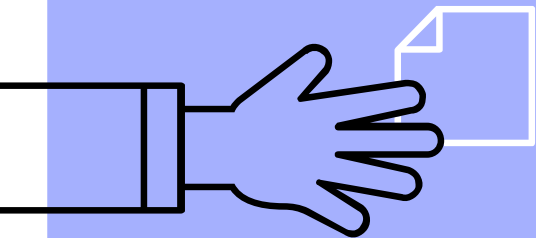
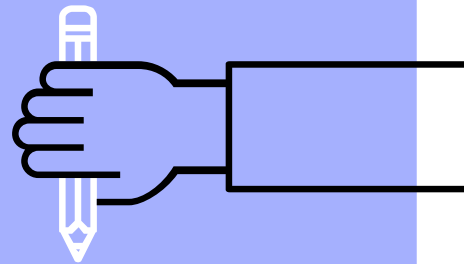
AlexNet foi a arquitetura de CNN escolhida por ser simples e eficiente. Foi o primeiro modelo usando técnicas de **Deep Learning** que ganhou o desafio da ImageNet batendo todo os outros algoritmos estatísticos.

Ferramentas

Como linguagem foi utilizado Python e as principais bibliotecas/frameworks utilizadas foram TFLearn, que é uma API em cima do TensorFlow e para processamento de imagem o OpenCV.



4. PÓS-PROCESSAMENTO



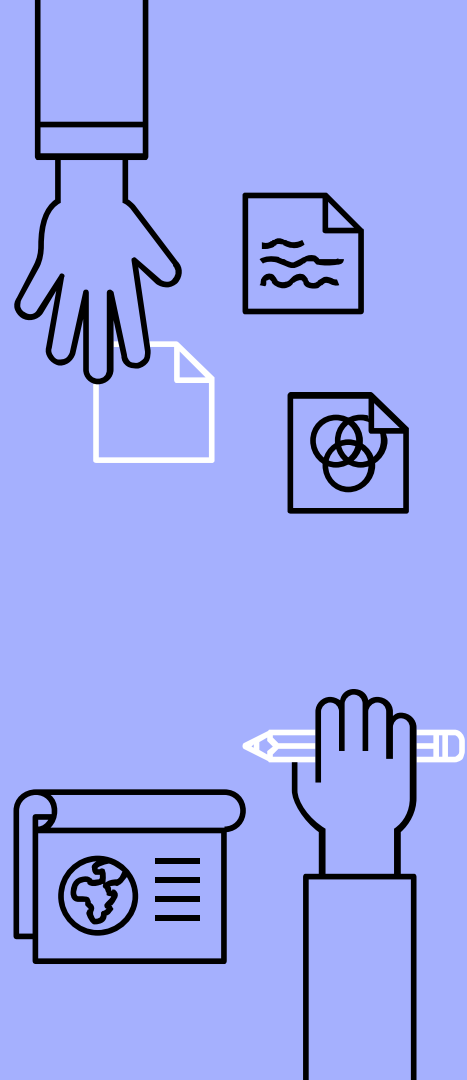
PÓS-PROCESSAMENTO

Coletamos mais dados sobre curvas.

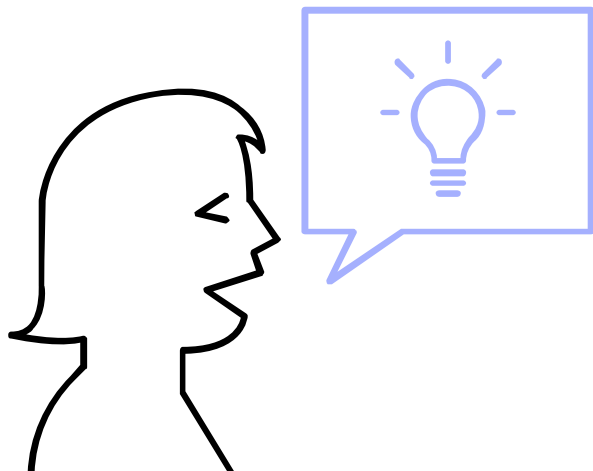
Os primeiros agentes **não sabiam muito bem fazer curvas**, pois havia pouca quantidade de dados com as direções esquerda e direita.

Mudança na lógica de tomada de decisão.

Fizemos algumas **mudanças na lógica** de tomada de decisão e o agente começou a dirigir melhor. Exemplo disso foi que, se ele andasse várias vezes seguidas para frente, decidimos **“mandar”** ele parar um pouco para **diminuir a velocidade**, pois o tempo de reação dele não é tão bom.



Dados do
último
treino.



+ - 50 MIL

Totais

Quantidade de dados

+ - 13 horas

Quantidade de horas treinadas

98%

Taxa de Acerto

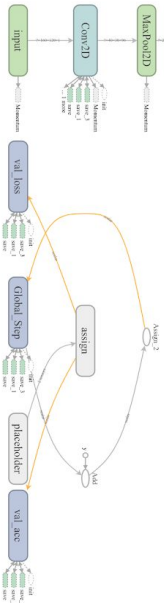
No Treinamento

89,8%

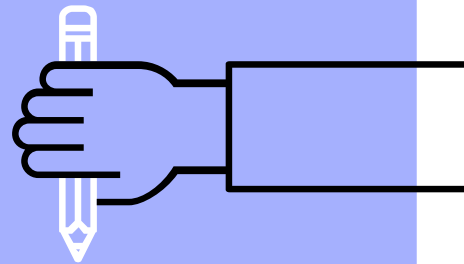
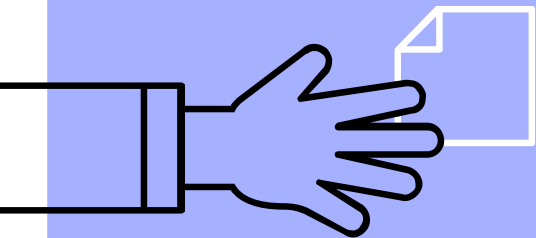
Taxa de Acerto

No Teste

14



5. UTILIZAÇÃO DO CONHECIMENTO

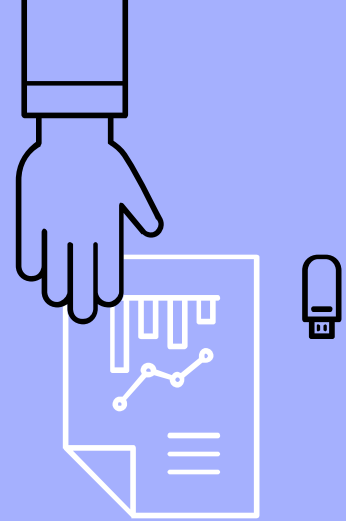


PROCESSO DE UTILIZAÇÃO

Capturar e
pré-processar
imagem do jogo

Fornecer a
imagem para o
modelo
treinado

Utilizar os
valores de
predição das
classes para
a lógica de
tomada de
decisão



OBRIGADO!

Alguma pergunta?

Integrantes:

Antonio Moreira	NºUSP: 9779242
Leonardo Meireles	NºUSP: 4182085
Luca Porto	NºUSP: 9778943
Vitor Brisola	NºUSP: 9791292

