Apresentação de Trabalho de Conclusão de Curso

Tema:

Análise comparativa de modelos de redes convolucionais na identificação de cromossomos em imagens digitais

Introdução

- Estabelecer os conceitos de Redes Neurais e Redes convolucionais
- Apresentar uma aplicação de uso eficiente que também contribua na área da biologia
- Apresentar a proposta de desenvolvimento e análise com base nos objetivos descritos
- Aplicar estruturas de redes convolucionais, destacando as diferenças e similaridades nos processos de treinamento
- Analisar os resultados obtidos
- Sugerir trabalhos futuros

Redes Neurais Artificiais

Nas palavras de HAYKIN (2001),

“Uma rede neural pode ser compreendida como um grande processador, maciçamente paralelo, constituído de inúmeros neurônios, que trocam informações e as reforçam para aprendizado entre si, através de sinapses(pesos) de conexão.”

Grande parte dos estudos iniciais do século XXI, relacionados a inteligência artificial , no escopo das redes neurais, foram difundidos pelo estudo de Simon Haykin, especialmente em seu livro, “***Neural Networks-A Comprehensive Foundation”***, publicado em 2001.

Por esse motivo, a maior parte dos conceitos teóricos deste trabalho foram baseados nas suas definições.

Estrutura do Neurônio Artificial 1/3

Figura 1

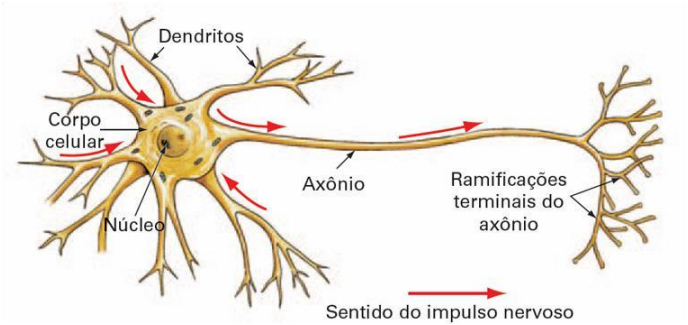
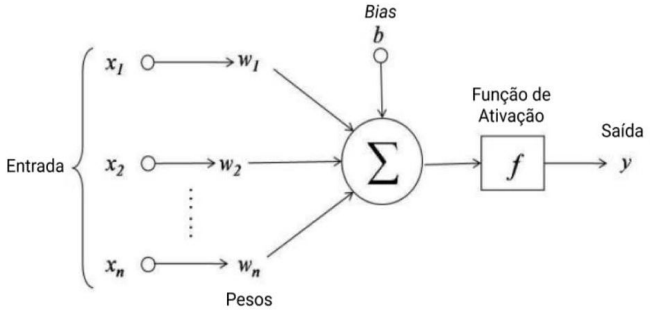


Figura 2



Na Figura 1, pode-se observar uma representação em cores da provável aparência de um neurônio biológico humano, enquanto que na Figura 2, é representado o modelo artificial de um neurônio, proposto por MCCULLOCH e PITTS (1943), chamado inicialmente de *Perceptron*.

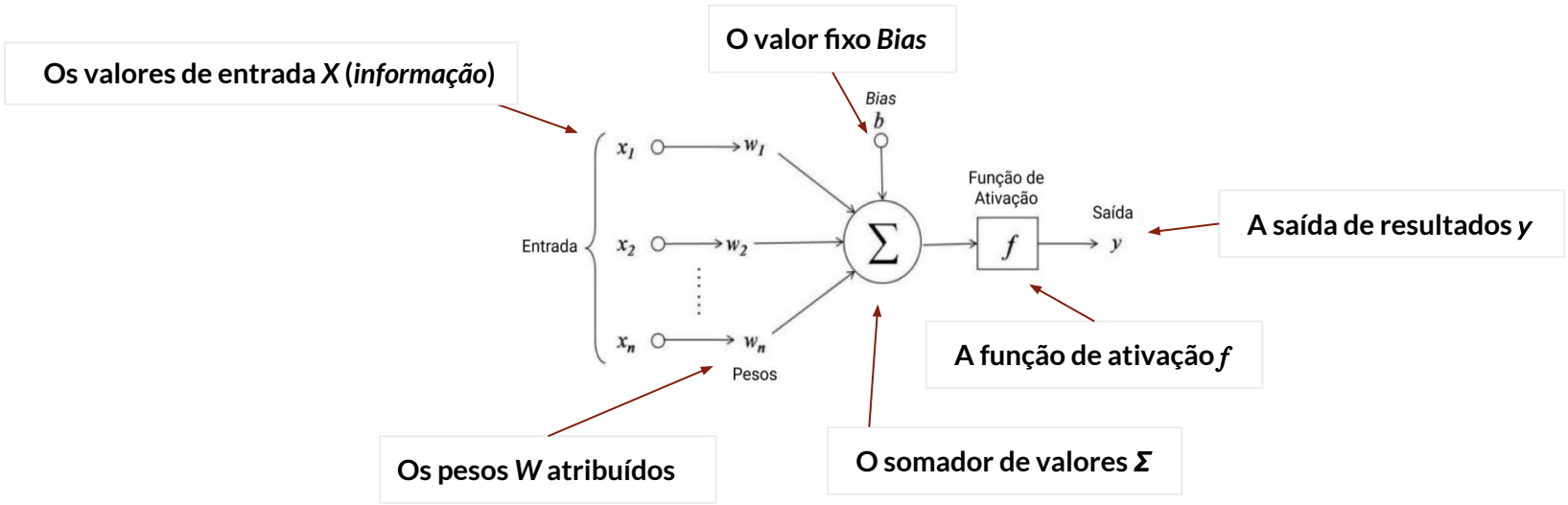
Estrutura do Neurônio Artificial 2/3

É possível observar uma semelhança entre ambas os “modelos”, isso porque, toda a concepção do neurônio artificial (Figura 2) foi baseada no modelo biológico (Figura 1), tanto na sua estrutura, como no seu funcionamento, principalmente no que diz respeito a forma de “transmissão” de informações, isso porque, conforme dito por HAYKIN (2001), a maior similaridade entre ambos está na forma que o aprendizado é feito, baseado no reforço.

Em um neurônio biológico, para que uma nova informação seja de fato “assimilada”, seu valor é reforçado entre os neurônios que compõem tal região uncubida desse processo, fazendo com que por persistência ela seja de fato assimilada. Já em um neurônio artificial, quando constituído em uma das várias camadas de uma RNA, as informações que o percorrem são atribuídas a um *peso*, que através do *algoritmo de aprendizagem* são corrigidos com acréscimos s ou decréscimos conforme a necessidade de resultado esperado do seu treinamento.

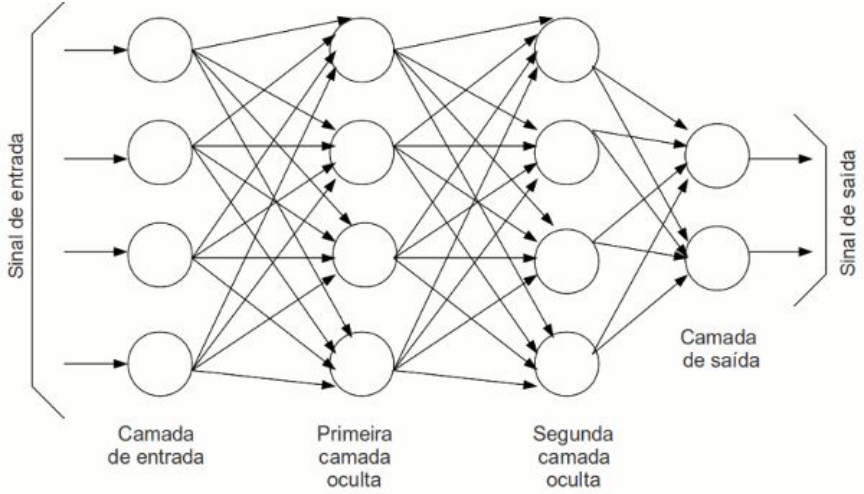
Estrutura do Neurônio Artificial 3/3

Basicamente, suas principais partes são:



Estrutura de uma RNA

Figura 4



Na Figura 4, temos a representação mais simples possível de uma estrutura de RNA, baseada no modelo de neurônio *Perceptron*.

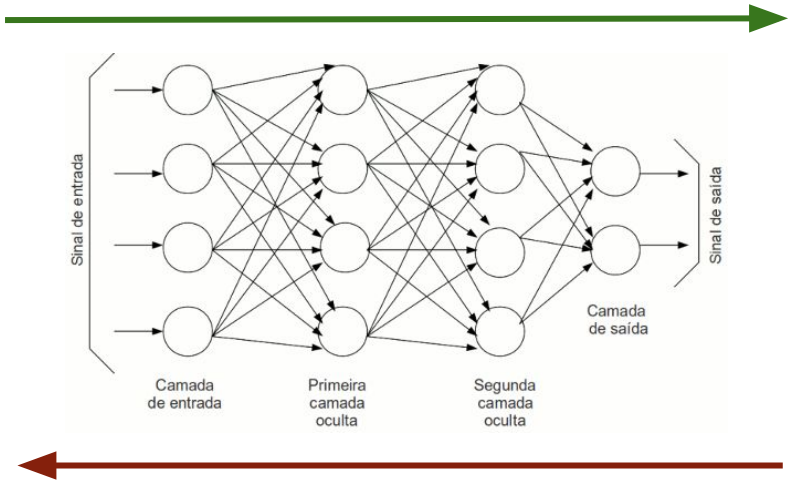
Seu funcionamento básico, se divide em camadas, partindo da **camada inicial**,onde cada neurônio que a compõe recebe um sinal de entrada, e o encaminha a todos os neurônios da **camada oculta** seguinte, que por sua vez transmitem as informações igualmente a **camada de saída**, que irá gerar os sinais de saída.

Pensando num treinamento supervisionado:

Quando obtidos, esses “resultados” de saída, eles serão comparados com os dados reais já previstos, sendo corrigidos seus pesos em uma *retropropagação* com algum *algoritmo de aprendizagem*.

Estrutura de uma RNA

Figura 5



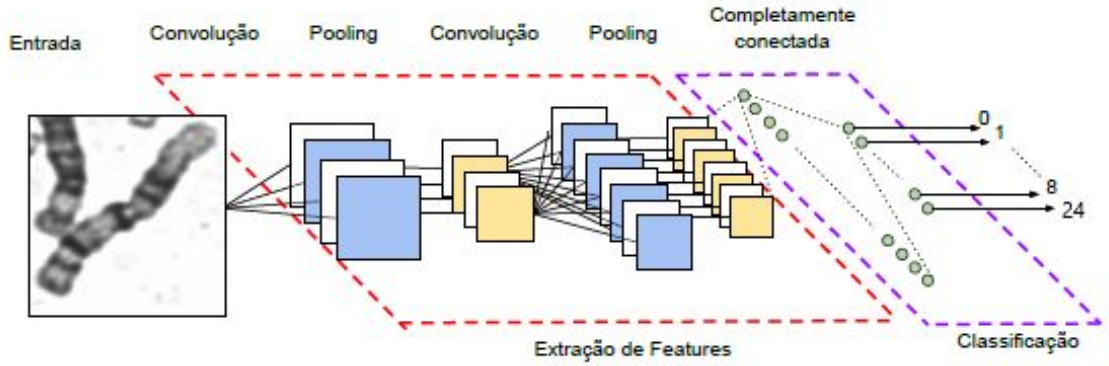
Essa correção dos *pesos*, força o reprocessamento dos dados através de todos os neurônios, para que o processo de treinamento seja refeito, esperando que os resultados de saída obtidos se aproximem ainda mais dos dados reais já conhecidos. Quando isso ocorre, é certo que os ajustes de aprendizado estão no caminho correto.

O algoritmo apresentado na Figura 5 é o *error backpropagation*, cuja operação consiste em passos de avanço e retrocesso, tendo a seta verde simulando o sentido do fluxo de informações, e a reta vermelha o fluxo de operação do algoritmo.

Redes Convolucionais

As redes convolucionais, ou *redes conexionistas* (CNN) partem muita da base das redes neurais artificiais, só que bem mais voltadas a técnicas de aprendizado profundo e visão computacional, sendo muito aplicadas a análises de imagens. Na Figura 6, é apresentada uma estrutura básica de uma CNN:

Figura 6



Camadas de uma CNN

Na definição das camadas base que definem um rede convolucional, temos:

- **Camada de convolução** : responsável por aplicar *filtros* ou *máscaras de convolução* sobre a imagem que será fornecida como dado de entrada a rede, produzindo os *mapas de características* ou *feature-maps*.

- **Camada de Pooling :** responsável por aplicar técnicas de redução em cima dos mapas obtidos anteriormente, como forma de sintetizar as características mais relevantes

- **Camadas totalmente conectadas:**
  - Camada de agrupamento: Fica incubida pelo agrupamento dos dados para classificação
  - Camada de classificação:Última camada presente na rede, responsável de fato pela classificação dos dados (classes).

Cromossomos

- O que é ?
  - 22 pares homólogos
  - 1 par sexuais
- Como foram classificados para o objetivo proposto.
- Suas classes:

Figura 7

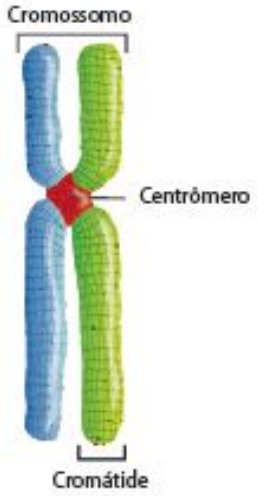
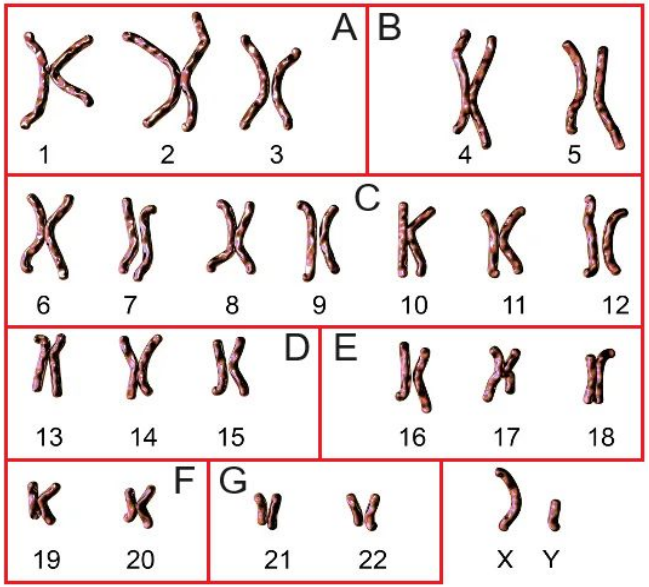


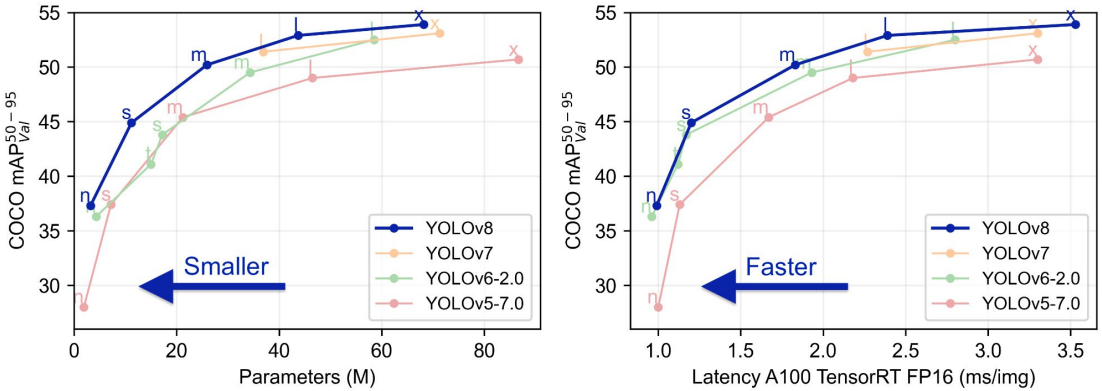
Figura 8



Modelo YoloV8

- Se diferencia dos sistemas de detecção que usam classificadores e localizadores;
- Aplica apenas uma rede neural à imagem completa;
- Divide a imagem em regiões e prevê as caixas delimitadoras e probabilidades de cada região;

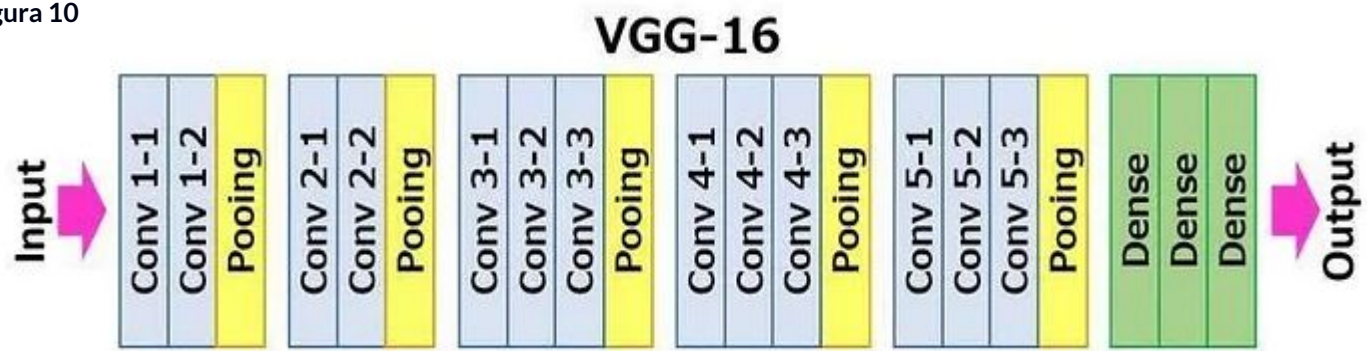
Figura 9



Modelo VGG16

- É um algoritmo de detecção e classificação de objetos, é um dos algoritmos populares para classificação;
- Usa um pequeno campo receptivo 3×3, os filtros 3×3 se combinam para fornecer a função de um campo receptivo maior;
- Utiliza um filtro convolucional menor, o que reduz a tendência da rede de se ajustar excessivamente durante os exercícios de treinamento;

Figura 10



Dataset

O dataset utilizado de base foi escolhido para o trabalho por possuir uma quantidade de 5000 imagens originais, obtidas em um ambiente laboratorial, apresentando uma boa qualidade para análise computacional. Outro fator ainda, é que as imagens já possuíam as atribuições dos cromossomos e de quais classes pertenciam, através de arquivos XML.

Figura 11

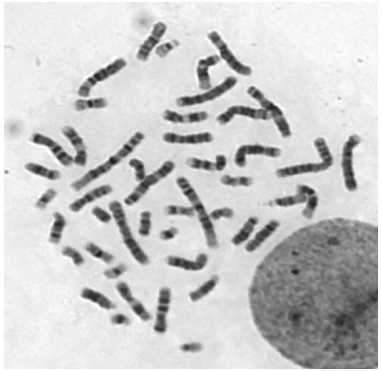
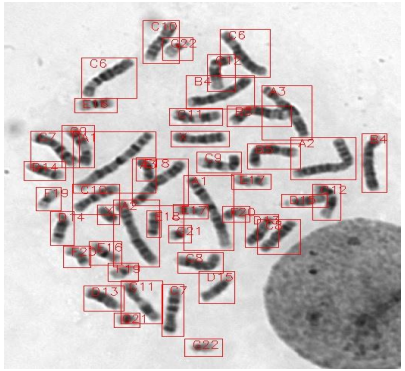


Figura 12



Desenvolvimento

- **YOLOV8**
  - Extrair dados dos arquivos de anotações XML;
  - Configurar arquivos e pastas de entrada e saída do modelo (pasta de treino e validação);
  - Configuração de anotações para formato YOLO (<class_id> <center_x> <center_y> <width> <height>);
  - Definição de hiperparâmetros como (dimensão das imagens, tamanho do lote e número de lotes);
  - Configurar ambiente de treinamento (GPU, CPU);
- **VGG16**
  - Extrair dados dos arquivos de anotações XML;
  - Configurar arquivos e pastas de entrada e saída do modelo (pasta de treino e validação);
  - Configuração de anotações para formato COCO JSON;
  - Definição de hiperparâmetros;
  - Configurar ambiente de treinamento CPU (Devido não ter suporte ao CUDA Nvidia);

Comparação entre Ambientes

Máquina 1:

- Intel Core I3 - 10ª Geração
- 4 Cores 8 Threads
- 16 GB RAM
- Nvidia GeForce GTX 1060 3 GB de VRAM
- Sistema Operacional: Windows 11

Máquina 2:

- AMD Ryzen 5 2400G
- 4 Cores 8 Threads
- 16 GB RAM
- AMD Radeon RX 580 8GB VRAM
- Sistema Operacional: Windows 10

Resultados obtidos do modelo YoloV8

- Treinamento relativo ao Yolo V8
- Resultados de treinamento no Ambiente de Máquina 1, visíveis na Tabela e grade de gráficos a partir da página 44
- Resultados de treinamento no Ambiente de Máquina 2, visíveis na Tabela e grade de gráficos a partir da página 45.

Tabela 4: Ambiente de execução em GPU - Máquina 1

Class	Images	Instances	Box Precision	Box Recall	Box mAP50	Box mAP50-95
Todas	1250	57487	0.981	0.967	0.992	0.979

Fonte:Autoria própria(2023)

Tabela 5: Ambiente de execução em CPU - Máquina 2

Class	Images	Instances	Box Precision	Box Recall	Box mAP50	Box mAP50-95
Todas	1250	57487	0.981	0.965	0.992	0.981

Fonte: Autoria própria (2023)

Resultados obtidos do modelo VGG16

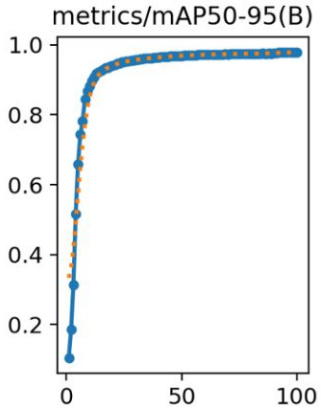
- Treinamento relativo ao VGG16 com os dados ajustados somente no ambiente de CPU da Máquina 1.
- Resultados de treinamento no Ambiente de Máquina 1, visíveis na grade de gráficos e na Tabela 5 a partir da página 52

Epoch	Time(seconds)	loss	accuracy	val_loss	val_accuracy
1	1156s	1.757	9.626	2.5210	1.0000
2	1115s	7.0878e-06	1.0000	6.8276e-07	1.0000
3	1150s	3.4014e-06	1.0000	2.1985e-07	1.0000
4	1089s	2.7047e-06	1.0000	6.8639e-08	1.0000
5	1121s	1.7634e-06	1.0000	7.6545e-09	1.0000
6	1121s	9.0196e-07	1.0000	1.2548e-10	1.0000

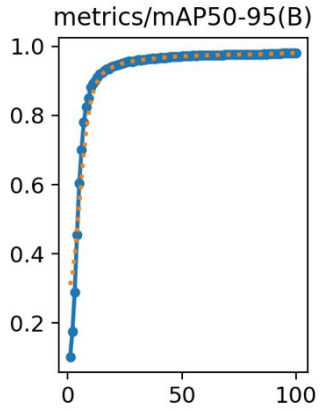
Conclusão 1/3

- **Modelo Yolo V8**
  - Comparação do desempenho entre máquina 1 e máquina 2
    - Tempo de execução da Máquina 1: 04 horas
    - Tempo de execução da Máquina 2: 56 horas.
  - Precisão média final entre as máquinas:
    - Precisão sobre a Máquina 1: 0.981 = 98%
    - Precisão sobre a Máquina 2: 0.981 = 98%

Máquina 1:



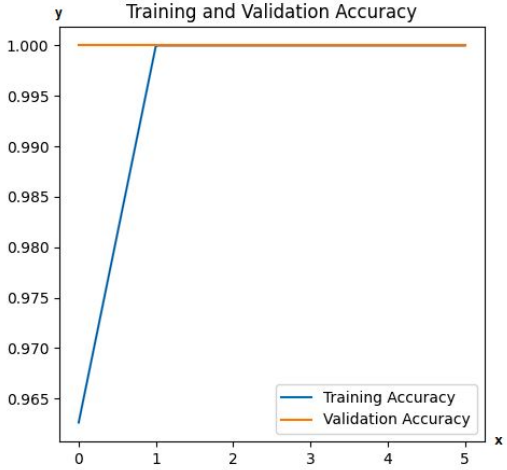
Máquina 2:

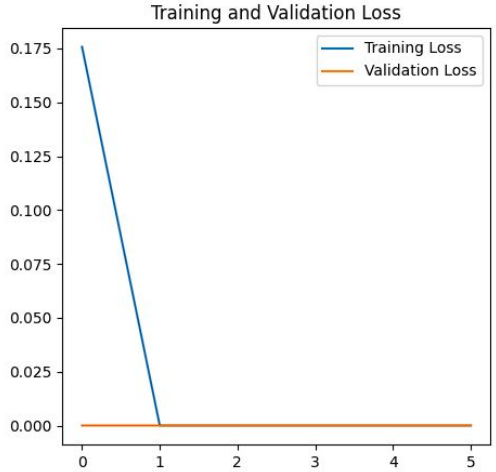



1050754.jpg
1051732.jpg
1050942.jpg
1052002.jpg
1052451.jpg
1052582.jpg

Conclusão 2/3

- **Modelo VGG16**
  - Pouca adaptabilidade relacionada ao processo na GPU
  - Valor de treinamento demonstrado como 100% a partir da 2 época.
  - Possível overfitting.





Conclusão 3/3

- Conclusão entre Yolo V8 x VGG16
  - Confiabilidade entre os modelos
  - Tempo gasto para o desenvolvimento
  - Precisão na identificação

- Considerações finais e propostas futuras
  - Em análise o objetivo final tem um desempenho mais adequado utilizando o YoloV8
  - Aprofundar o conhecimento em cima de algoritmos DeepLearning para uma análise reestruturada dos modelos.

Agradecimentos!

Orientador: Paulo César Polastri

Nathan Lucas Santos Nicolau - F310FH5

Márcio Rogério Spadari Júnior - N6110C0

Matheus Pereira da Silva - N610CH2