

# **Relatório Técnico da Disciplina de**

## **Tópicos Especiais em Inteligência Artificial**

Aluno: Lucca Souza Di Oliveira  
RGA: 202011310024

### **1. Introdução**

O projeto apresenta uma Inteligência Artificial que é fruto da combinação entre deep learning e aprendizado de máquinas (machine learning) para resolver o desafio de identificar e classificar imagens de pássaros, assim facilitando e auxiliando aqueles que podem aplicar dessa tecnologia de alguma forma, assim como no artigo “Deep Learning Case Study for Automatic Bird Identification” de Juha Niemi & Juha T. Tantt, o qual utiliza do recurso para auxiliar em parques eólicos na Filândia, ou até mesmo tais que levam a ornitologia como interesse, tal qual no artigo “Bird Image Retrieval and Recognition Using a Deep Learning Platform” de Yo-Ping Huang e Haobijam Basanta. Ao empregar redes neurais profundas treinadas em grandes conjuntos de dados de imagens de pássaros, a IA conseguiu extrair os recursos complexos e sutis que caracterizam diferentes espécies. Isso permite que a IA analise e compare características importantes, como forma, tamanho, cor e padrão de penas, para classificar com precisão os pássaros nas fotos e fornecer os nomes correspondentes de forma imediata.

Há benefícios tanto para ornitólogos amadores até pesquisadores científicos. A solução irá ajudar a catalogar com maior precisão a biodiversidade de aves e, também, a desenvolver estratégias de auxílio para problemas envolvendo identificação de pássaros. Além disso, a inteligência artificial pode ser integrada a dispositivos móveis com câmeras com o objetivo de educar e envolver o público na apreciação e monitoramento de aves, inspirando uma maior consciência ambiental.

### **2. Metodologia**

O dataset foi escolhido de uma base de dados disponível no kaggle (<https://www.kaggle.com/datasets/gpiosenka/100-bird-species>), porém houve modificações manuais para selecionar os pássaros brasileiros que tinham na base da dados utilizada, dado que foi encontrado em “<https://www.birdlist.org/brazil.htm>”. Após a seleção, foi-se feito uma análise dos pássaros restantes e houve exclusão dos pássaros típicos da região norte e nordeste após conclusão de que tais animais têm uma taxa mais baixa de migração para a região Centro-Oeste, território qual foi escolhida como alvo do projeto. Dessa forma, concluiu-se 62 classes as quais foram formadas das espécies “AMERICAN KESTREL”, 'AMERICAN REDSTART', 'AMETHYST WOODSTAR',

'ANHINGA', 'BAR-TAILED GODWIT', 'BARN OWL', 'BARN SWALLOW', 'BEARDED BELLBIRD', 'BLACK NECKED STILT', 'BLACK SKIMMER', 'BLACK VULTURE', 'BLACKBURNIAM WARBLER', 'BLONDE CRESTED WOODPECKER', 'BOBOLINK', 'CAMPO FLICKER', 'CAPPED HERON', 'CAPUCHINBIRD', 'CINNAMON ATTILA', 'CINNAMON TEAL', 'CRANE HAWK', 'CREAM COLORED WOODPECKER', 'EASTERN MEADOWLARK', 'GREAT JACAMAR', 'GREAT KISKADEE', 'GREAT POTOO', 'GREAT XENOPS', 'GREY PLOVER', 'HARPY EAGLE', 'HORNED SUNGEM', 'HOUSE SPARROW', 'IVORY BILLED ARACARI', 'KING VULTURE', 'LAUGHING GULL', 'LIMPIN', 'MANGROVE CUCKOO', 'ORNATE HAWK EAGLE', 'OSPREY', 'PEREGRINE FALCON', 'POMARINE JAEGER', 'PURPLE GALLINULE', 'PURPLE MARTIN', 'RED BILLED TROPICBIRD', 'RED KNOT', 'ROCK DOVE', 'ROSEATE SPOONBILL', 'RUFUOS MOTMOT', 'SAND MARTIN', 'SCARLET MACAW', 'SHORT BILLED DOWITCHER', 'SNOWY EGRET', 'SNOWY SHEATHBILL', 'SPANGLED COTINGA', 'SQUACCO HERON', 'STRIPED OWL', 'STRIPPED MANAKIN', 'SUNBITTERN', 'TROPICAL KINGBIRD', 'TURKEY VULTURE', 'VEERY', 'WATTLED CURASSOW', 'WHIMBREL', 'WOOD THRUSH' “.

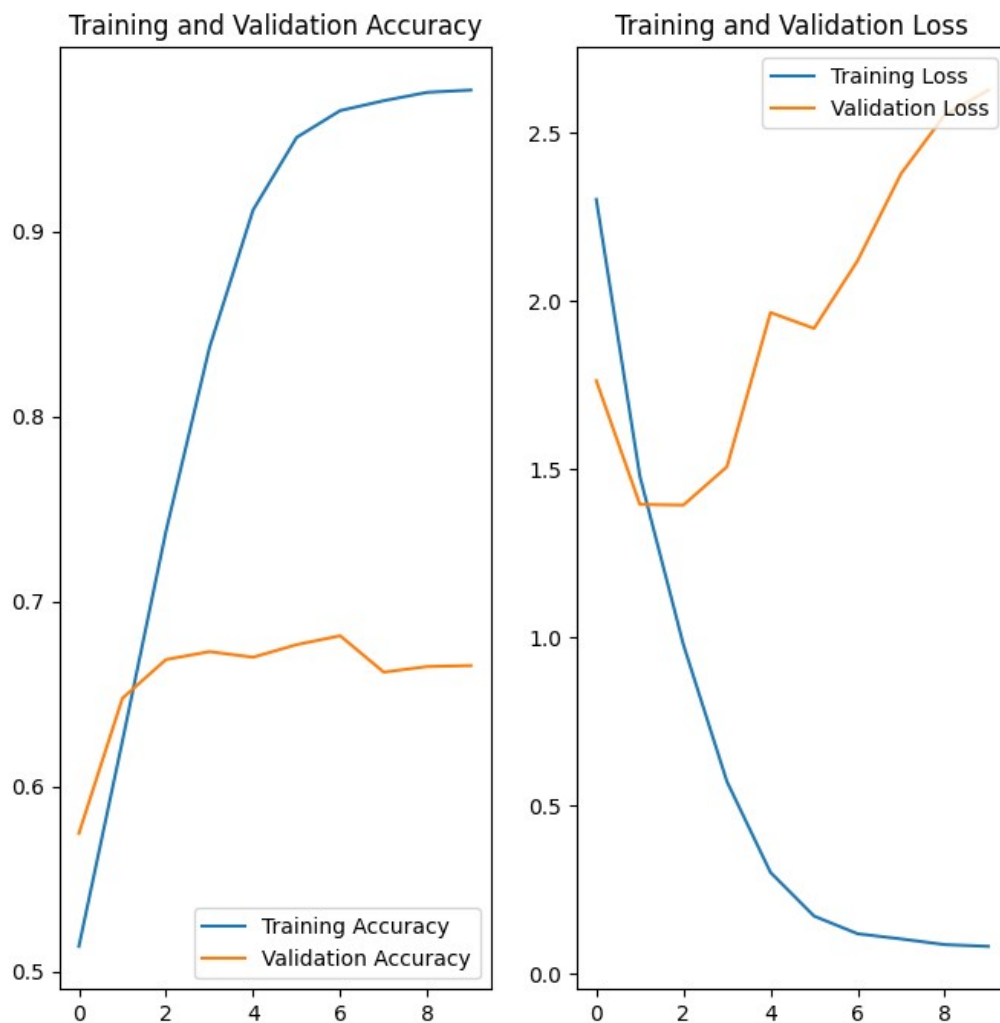
## 2.1. Treinamento

Foi-se separados em treinamento e teste nas proporções de 80% treinamento e 20% de teste. Assim utilizando 15892 imagens para treino e 3972 para teste, totalizando 19864 imagen, com uma média aproximada de 150 imagens por classe.

### 2.1.2. Épocas

Houve uma divião em dez épocas, sendo que a maior acurácia atingida foi de 68% sem a utilização de Data Augmentation.

```
Epoch 1/10
497/497 [=====] - 138s 276ms/step - loss: 2.3027 - accuracy: 0.5137 - val_loss: 1.7634 - val_accuracy: 0.5748
Epoch 2/10
497/497 [=====] - 134s 270ms/step - loss: 1.4798 - accuracy: 0.6247 - val_loss: 1.3955 - val_accuracy: 0.6478
Epoch 3/10
497/497 [=====] - 134s 271ms/step - loss: 0.9784 - accuracy: 0.7380 - val_loss: 1.3934 - val_accuracy: 0.6687
Epoch 4/10
497/497 [=====] - 134s 269ms/step - loss: 0.5711 - accuracy: 0.8374 - val_loss: 1.5074 - val_accuracy: 0.6730
Epoch 5/10
497/497 [=====] - 134s 269ms/step - loss: 0.3008 - accuracy: 0.9116 - val_loss: 1.9660 - val_accuracy: 0.6699
Epoch 6/10
497/497 [=====] - 135s 272ms/step - loss: 0.1710 - accuracy: 0.9508 - val_loss: 1.9189 - val_accuracy: 0.6767
Epoch 7/10
497/497 [=====] - 134s 270ms/step - loss: 0.1184 - accuracy: 0.9653 - val_loss: 2.1203 - val_accuracy: 0.6815
Epoch 8/10
497/497 [=====] - 134s 270ms/step - loss: 0.1030 - accuracy: 0.9706 - val_loss: 2.3796 - val_accuracy: 0.6619
Epoch 9/10
497/497 [=====] - 135s 271ms/step - loss: 0.0861 - accuracy: 0.9751 - val_loss: 2.5535 - val_accuracy: 0.6649
Epoch 10/10
497/497 [=====] - 135s 271ms/step - loss: 0.0809 - accuracy: 0.9763 - val_loss: 2.6277 - val_accuracy: 0.6654
```



## 2.2. Data Augmentation

Para aumentar a diferença entre as imagens, foi-se utilizado a técnica de Data Augmentation. Assim, foram alterados os seguintes parâmetros das imagens: rotatividade; zoom; flip da esquerda para direita; flip de cima para baixo e inclinação da imagem. Diante da execução, foram criadas, um total de, 19.964 imagens, assim tendo uma média aproximada de 300 imagens por classe.

Espécie	Teste	Treino
American Kestel	5	314
American redstart	5	280
Amethyst woodstar	5	263
Anhinga	5	314
Barn Swallow	5	321
Barn Owl	5	269
Bar-tailed Godwit	5	339
Bearded Bellbird	5	289

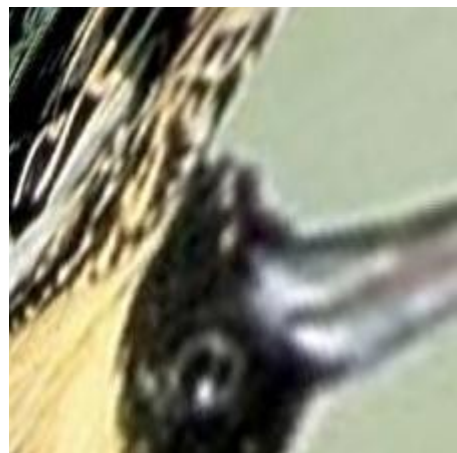
BLACK NECKED STILT	5	349
Black skimmer	5	316
Black vulture	5	319
Blackburnian Warbler	5	273
BLONDE CRESTED	5	291
WOODPECKER		
Bobolink	5	319
Campo Flicker	5	400
Capped Heron	5	278
Capuchinbird	5	268
Cinnamon Attila	5	331
Cinnamon Teal	5	325
Crane Hawk	5	274
Cream Colored Woodpecker	5	280
Eastern Meadowlark	5	385
Great Jacamar	5	335
Great Kiskadee	5	342
Great Potoo	5	278
Great Xenops	5	306
Grey Plover	5	320
Harpy Eagle	5	353
Horned Sungem	5	299
House Sparrow	5	314
Ivory Billed Aracari	5	301
King Vulture	5	275
Laughing Gull	5	418
Limpkin	5	406
Mangrove Cuckoo	5	276
Ornate Hawk Eagle	5	424
Osprey	5	310
Peregrine Falcon	5	321
Pomarine Jaeger	5	287
Purple Gallinule	5	310
Purple Martin	5	323
Red Billed Tropicbird	5	427
Red Knot	5	408
Rock Dove	5	266
Roseate Spoonbill	5	393
Rufous MotMot	5	383
Sand Martin	5	306
Scarlet Macaw	5	329
Short Billed Dowitcher	5	337
Snowy Egret	5	265
Snowy Sheathbill	5	377
Spangled Cotinga	5	326
Squacco Heron	5	307
Striped Owl	5	288
Stripped Manakin	5	291
Sunbitern	5	345
Tropical Kingbird	5	310
Turkey Vulture	5	329
Veery	5	391

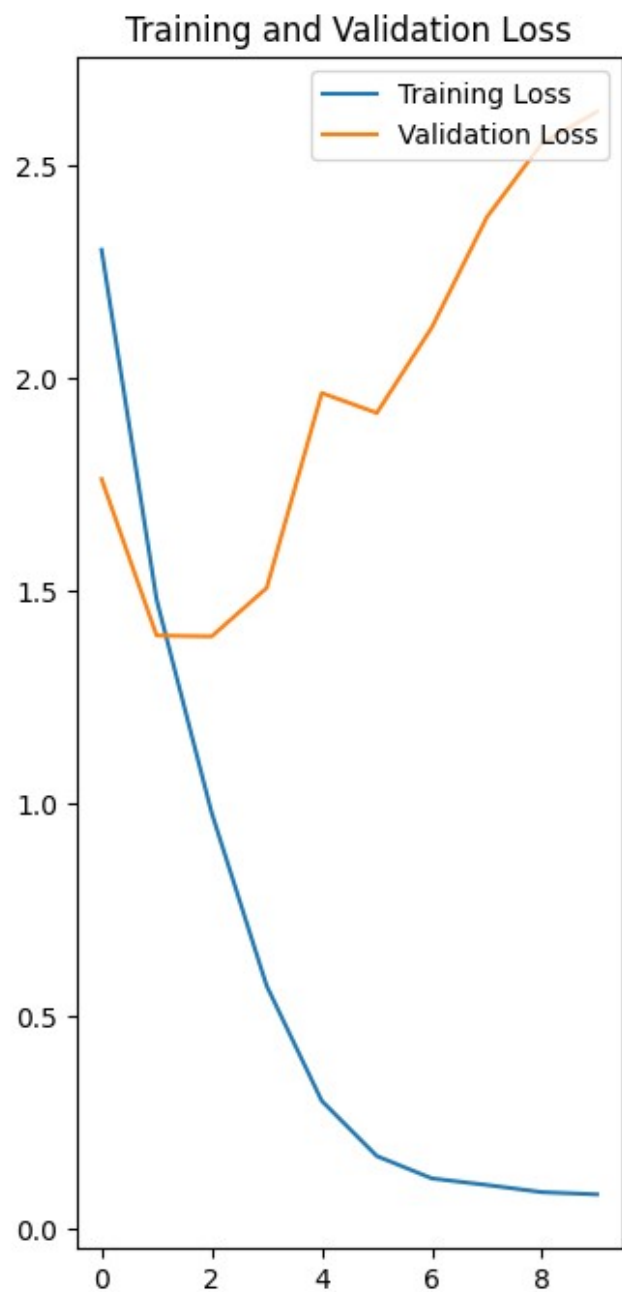
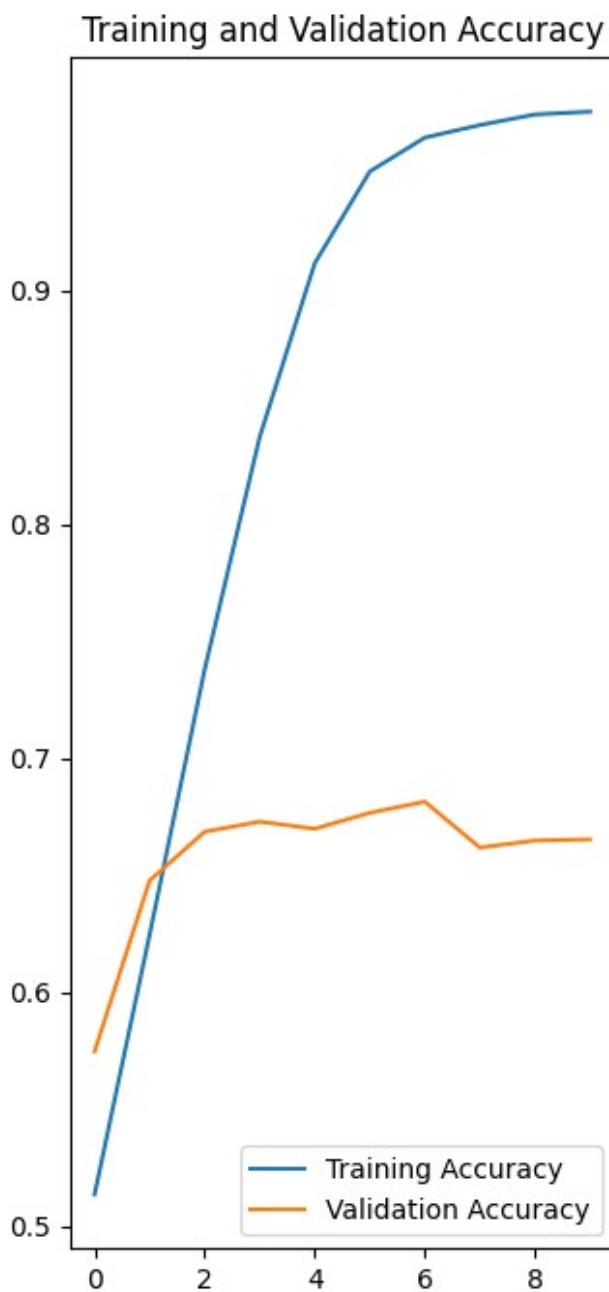
Wattled Curassow	5	277
Whimbrel	5	282
Wood Thush	5	432
Total	310	19496

### 2.2.3. Resultado da Augmentação.

A augmentação foi feita em duas etapas, no qual a segunda obteve oito épocas e uma acurácia de aproximadamente 69%.

```
Epoch 1/8
502/502 [=====] - 131s 261ms/step - loss: 3.5619 - accuracy: 0.1085 - val_loss: 3.2758 - val_accuracy: 0.1620
Epoch 2/8
502/502 [=====] - 126s 252ms/step - loss: 3.0896 - accuracy: 0.2000 - val_loss: 3.1004 - val_accuracy: 0.1907
Epoch 3/8
502/502 [=====] - 125s 250ms/step - loss: 2.7061 - accuracy: 0.2849 - val_loss: 3.1912 - val_accuracy: 0.2011
Epoch 4/8
502/502 [=====] - 126s 251ms/step - loss: 2.2159 - accuracy: 0.4057 - val_loss: 3.4014 - val_accuracy: 0.1944
Epoch 5/8
502/502 [=====] - 127s 252ms/step - loss: 1.6956 - accuracy: 0.5383 - val_loss: 4.0746 - val_accuracy: 0.1892
Epoch 6/8
502/502 [=====] - 128s 255ms/step - loss: 1.2690 - accuracy: 0.6552 - val_loss: 5.0899 - val_accuracy: 0.1732
Epoch 7/8
502/502 [=====] - 127s 252ms/step - loss: 0.9701 - accuracy: 0.7322 - val_loss: 5.7100 - val_accuracy: 0.1672
Epoch 8/8
502/502 [=====] - 128s 254ms/step - loss: 0.7495 - accuracy: 0.7914 - val_loss: 6.6478 - val_accuracy: 0.1742
```





## Tecnologias.

### Ambiente de Desenvolvimento

#### 1. Jupyter Notebook

O ambiente de programação Jupyter, junto com a linguagem de programação Python, desempenhou um papel fundamental neste projeto. Foram utilizados notebooks Jupyter para criar, executar e

documentar o código relacionado à inteligência artificial identificadora de pássaros. Isso incluiu a implementação de algoritmos de aprendizado de máquina e análise de dados exploratórios.

## 2. Hardware

O hardware empregado neste projeto consistia nos seguintes componentes:

- **CPU:** AMD Ryzen 7 5800X com 8 núcleos e 16 threads, oferecendo alto desempenho computacional para treinamento de modelos de IA e processamento intensivo.
- **GPU:** Nvidia RTX 3060 Ti, uma GPU de alto desempenho com núcleos CUDA, projetada para cargas de trabalho de aprendizado de máquina. A GPU acelerou significativamente os cálculos relacionados à IA, permitindo treinamentos mais rápidos e eficientes de modelos de redes neurais.
- **Memória RAM:** 32 GB de RAM DDR4, proporcionando largura de banda suficiente para lidar com grandes volumes de dados e modelos de IA complexos.

## Sistema Operacional

O sistema operacional escolhido para este projeto foi o Windows 10. O que se encaixou muito bem devido à sua ampla compatibilidade com software de desenvolvimento e ferramentas relacionadas à IA. Além disso, ofereceu um ambiente familiar para os desenvolvedores e cientistas de dados envolvidos no projeto.

## Implementação de IA

A implementação de IA neste projeto abrangeu várias áreas, incluindo:

- **Treinamento de Modelos de Aprendizado de Máquina:** Utilizando bibliotecas como TensorFlow e Keras, modelos de redes neurais foram treinados para tarefas de classificação, regressão e processamento de imagem.
- **Análise de Dados:** NumPy e Matplotlib foram usadas como bibliotecas de análise de dados e foram empregadas para explorar e visualizar dados, fornecendo insights valiosos para o projeto.
- **Uso da GPU para aceleração dos resultados:** Foi-se necessário a instalação de diversos softwares gráficos, como o CUDA v11.2 e o cuDNN v8.1, além de ser, também, necessário o uso de versões compatíveis, como o TensorFlow-gpu-2.10.0 e o Python 3.10.

## Transfer Learning.

Nesta seção da documentação, discutiremos a implementação de Transfer Learning usando três modelos pré-treinados populares do Keras: ResNet50V2, MobileNetV2 e InceptionV3. O Transfer Learning é uma técnica que permite aproveitar modelos de deep learning pré-treinados em tarefas específicas, economizando tempo e recursos de treinamento.

### Congelamento de Camadas

Em alguns treinamentos específicos, foi-se optado por congelar apenas uma parte das camadas do modelo pré-treinado, em vez de congelar todas as camadas. Isso permite ajustar o modelo para tarefas mais complexas ou para conjuntos de dados maiores. O que pode congelar uma porcentagem das camadas do modelo pré-treinado, como 40% ou 80%, dependendo das características da tarefa.

### Tecnologia utilizadas

- 1. **ResNet50V2:** Uma versão da ResNet (Rede Neural Residual) com 50 camadas. É conhecida por sua capacidade de treinar redes profundas com estabilidade.
- 2. **MobileNetV2:** Uma arquitetura de rede neural convolucional otimizada para aplicações móveis e embarcadas. É eficiente em termos de recursos computacionais e adequada para dispositivos com recursos limitados.
- 3. **InceptionV3:** Uma rede neural profunda que utiliza a ideia de convoluções multi-nível para extrair características em diferentes escalas espaciais. É conhecida por seu desempenho em tarefas de classificação de imagens.

### Tabela de acurácia do Transfer learning

Modelos	Precisão	Épocas	freeze
InceptionV3	30.902%	10	40%
InceptionV3	30.902%	10	80%
MobileNetV2	25.694%	10	40%
MobileNetV2	25.694%	10	80%
ResNet50V2	40.625%	10	40%
ResNet50V2	25.694%	10	80%



## Discussão

Durante a implementação do Transfer Learning com os modelos pré-treinados ResNet50V2, MobileNetV2 e InceptionV3, encontramos desafios significativos relacionados ao tamanho do conjunto de dados e à necessidade de acelerar o treinamento mudando para a GPU.

### Tamanho do Conjunto de Dados

Um dos principais desafios que enfrentamos foi o tamanho do conjunto de dados. O sucesso do Transfer Learning depende de ter um conjunto de dados de destino representativo e significativo para a tarefa em questão. No entanto, em algumas situações, como projetos com recursos limitados, pode ser difícil reunir um grande conjunto de dados de treinamento.

Nossa base de dados de destino era relativamente pequena em comparação com os modelos pré-treinados que usamos. Isso pode resultar em problemas de overfitting, onde o modelo se ajusta demais aos dados de treinamento, mas não generaliza bem para novos dados. Para combater esse problema, aplicamos técnicas de regularização e consideramos congelar uma parte das camadas do modelo pré-treinado, como mencionado anteriormente.

### Mudança para a GPU

O treinamento de modelos de deep learning em conjuntos de dados grandes e modelos complexos pode ser computacionalmente intensivo. O uso de uma GPU (Unidade de Processamento Gráfico) pode acelerar significativamente o treinamento, permitindo que os cálculos sejam realizados de forma paralela. No entanto, a mudança para a GPU pode ser um processo desafiador, especialmente se você não estiver familiarizado com a configuração e o uso de hardware de GPU.

Durante o desenvolvimento, teve-se que aprender a configurar adequadamente a GPU, instalar os drivers apropriados (CUDA e cudnn) e garantir que nosso ambiente de desenvolvimento estivesse configurado corretamente para aproveitar o poder computacional da GPU. Isso envolveu um investimento de tempo e esforço, mas foi essencial para acelerar o treinamento e obter resultados mais rápidos.

## Conclusão

A implementação bem-sucedida do Transfer Learning com os modelos ResNet50V2, MobileNetV2 e InceptionV3 demonstrou ser uma estratégia eficaz para tarefas de Deep Learning. No entanto, foi-se enfrentado desafios notáveis relacionados ao tamanho do conjunto de dados de destino e à necessidade de melhorar o tempo para treinamento em GPU.

A dificuldade de treinar a base devido ao tamanho do conjunto de dados foi mitigada por meio de técnicas de regularização, congelamento de camadas e ajustes cuidadosos dos hiperparâmetros do modelo. Isso permitiu que o modelo pré-treinado fosse adaptado de forma para a tarefa específica.

Além disso, a mudança para a GPU foi fundamental para acelerar o treinamento e reduzir significativamente o tempo necessário para obter resultados. Embora tenha havido desafios na configuração inicial, a capacidade de utilizar hardware de GPU foi uma vantagem crítica para treinar modelos em um tempo razoável.

## Bibliografia

Niemi, J.; Tantt, J.T. Deep Learning Case Study for Automatic Bird Identification. Appl. Sci. 2018, 8, 2089. <https://doi.org/10.3390/app8112089>

Y. -P. Huang and H. Basanta, "Bird Image Retrieval and Recognition Using a Deep Learning Platform," in *IEEE Access*, vol. 7, pp. 66980-66989, 2019, doi: 10.1109/ACCESS.2019.2918274.