

# Project 1: Deep Learning-Image Classification with CNN

Group 3

**Presented by**

Mauricio

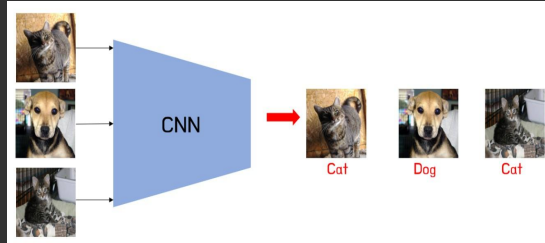
Nicolas

# How the partnership will work



## Introduction

The main goal of this project is to develop an image classification model capable of correctly identifying 10 different animal species.



## Data set

The Animals-10 dataset used in this project contains approximately 28,000 images distributed across 10 different animal classes: dog, cat, horse, elephant, butterfly, chicken, cow, sheep, spider, and squirrel.

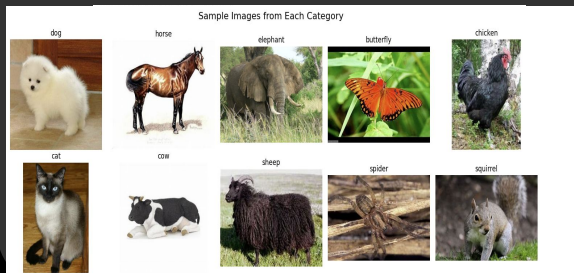
```
import kagglehub

# Download latest version
path = kagglehub.dataset_download("alessiocrorad  
99/animals10")

print("Path to dataset files:", path)
```

## Visualization / Analysis

1



## Preprocessing

2

### CONFIGURATION PARAMETERS:

TARGET\_SIZE: (224, 224)

BATCH\_SIZE: 64

VALIDATION\_SPLIT: 0.2

SEED: 42

EPOCHS: 25

LEARNING\_RATE: 0.001

MODEL\_PATH: /content/drive/MyDrive/animals\_project/models/best\_model.h5

## Augmentation

3

```
train_datagen = ImageDataGenerator(  
    preprocessing_function=preprocess_input,  
    rotation_range=15,  
    width_shift_range=0.15,  
    height_shift_range=0.15,  
    shear_range=0.15,  
    zoom_range=0.15,  
    brightness_range=[0.8, 1.2],  
    horizontal_flip=True,  
    fill_mode='nearest',  
    validation_split=VALIDATION_SPLIT  
)
```

## Validation Split

4

- 80% for training
- 20% for testing



# MODELS AND TESTING

**Architecture:** 3 Conv+Pooling blocks → Flatten → Dense(256)+Dropout → Output  
**Filters:** 32 → 64 → 1180 (aggressive expansion)  
**Activation:** ReLU (hidden), Softmax (output)  
**Regularization:** 50% Dropout  
**Optimization:** Adam (lr=0.0005) + Categorical Crossentropy  
**I/O:** 180x180x3 input → 10-class output  
**Parameters:** ~8.5M  
**Performance:** ~60% Accuracy

60%




**Architecture:** 3 Conv+Pooling blocks → Flatten → Dense(256)+Dropout → Output  
**Filters:** 32 → 64 → 1180 (aggressive expansion)  
**Activation:** ReLU (hidden), Softmax (output)  
**Regularization:** 50% Dropout  
**Optimization:** Adam (lr=0.0002) + Categorical Crossentropy  
**I/O:** 224x224x3 input → 10-class output  
**Parameters:** ~8.5M  
**Performance:** ~80% Accuracy

80%

**Architecture:** 4 Conv+Pooling blocks → Flatten → Dense+Dropout → Output  
**Filters:** 32 → 64 → 128 → 128 (progressive increase)  
**Activation:** ReLU (hidden), Softmax (output)  
**Regularization:** 50% Dropout  
**Optimization:** Adam + Categorical Crossentropy  
**I/O:** 224x224x3 input → 10-class output

75%

## Transfer Learning

-  EfficientNetB0
-  Resnet50
-  MobileNetV2

## Fine - Tuning

```
# Unfreeze top layers for fine-tuning (keep early layers frozen)
FINE_TUNE_AT = 100 # For GPU L4, this is a safe unfreeze depth
for layer in model_tl.layers[:FINE_TUNE_AT]:
    layer.trainable = False
for layer in model_tl.layers[FINE_TUNE_AT:]:
    layer.trainable = True
```

98%

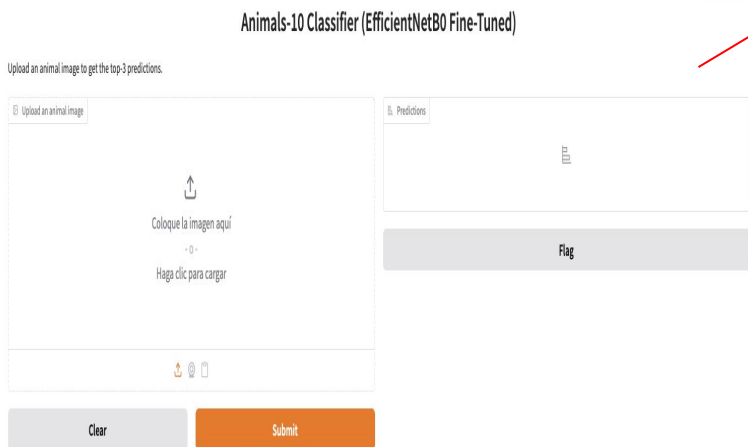
### VALIDATION RESULTS:

- Validation Accuracy: 0.9784
- Validation Loss: 0.0722

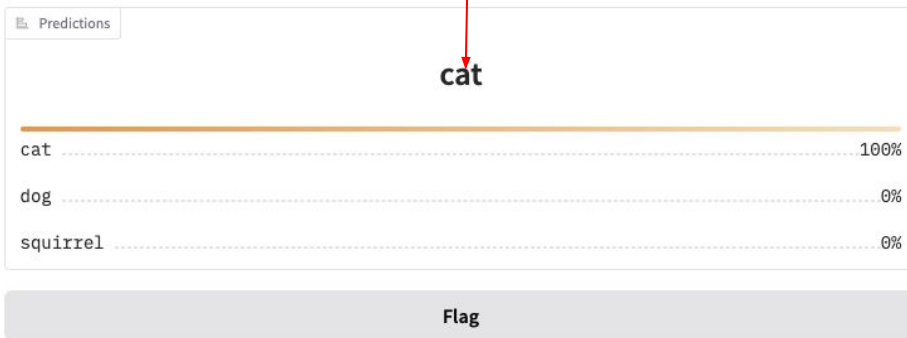
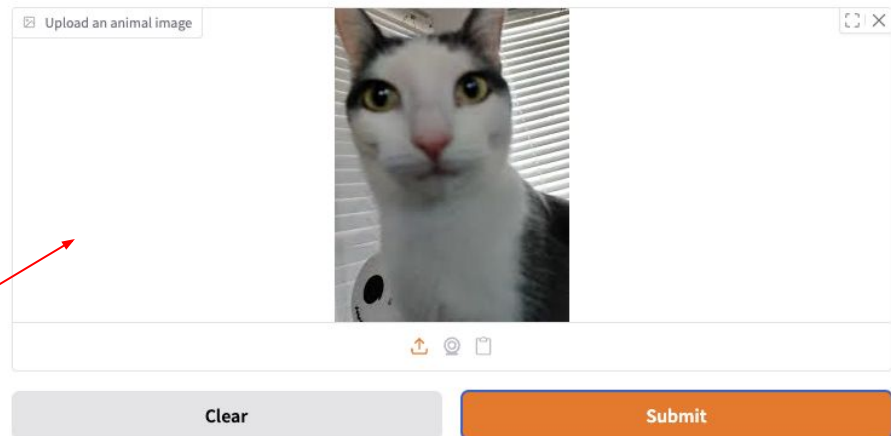
Let's work together

# Deployment

- We use Gradio for the deployment.



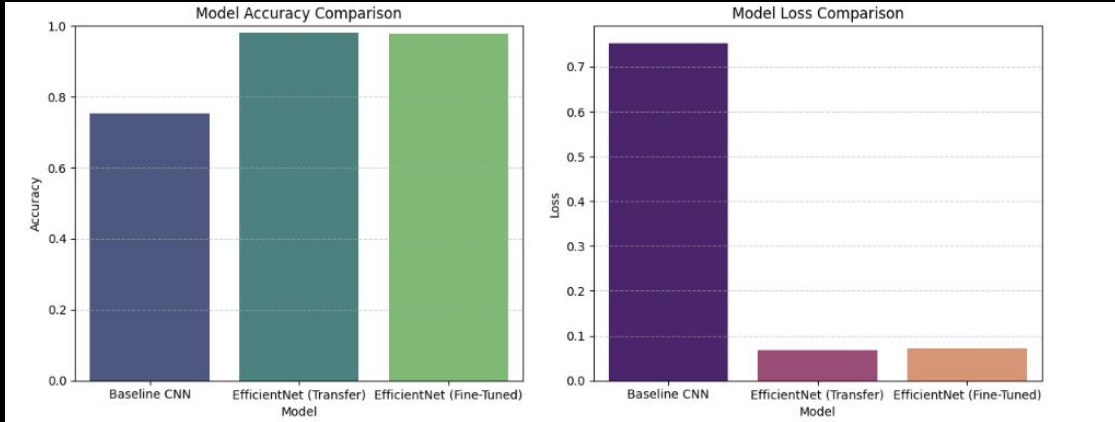
Upload an animal image to get the top-3 predictions.



# RESULTS

## VALIDATION RESULTS:

- Validation Accuracy: 0.9784
- Validation Loss: 0.0722



# RESULTS

## CLASSIFICATION REPORT:

	precision	recall	f1-score	support
cane	0.99	0.98	0.98	972
cavallo	0.97	0.98	0.97	524
elefante	0.95	0.99	0.97	289
farfalla	0.98	0.99	0.98	422
gallina	0.99	0.99	0.99	619
gatto	0.98	0.96	0.97	333
mucca	0.93	0.97	0.95	373
pecora	0.97	0.94	0.95	364
ragno	0.99	0.99	0.99	964
scoiattolo	0.98	0.97	0.98	372
accuracy			0.98	5232
macro avg	0.97	0.98	0.97	5232
weighted avg	0.98	0.98	0.98	5232

