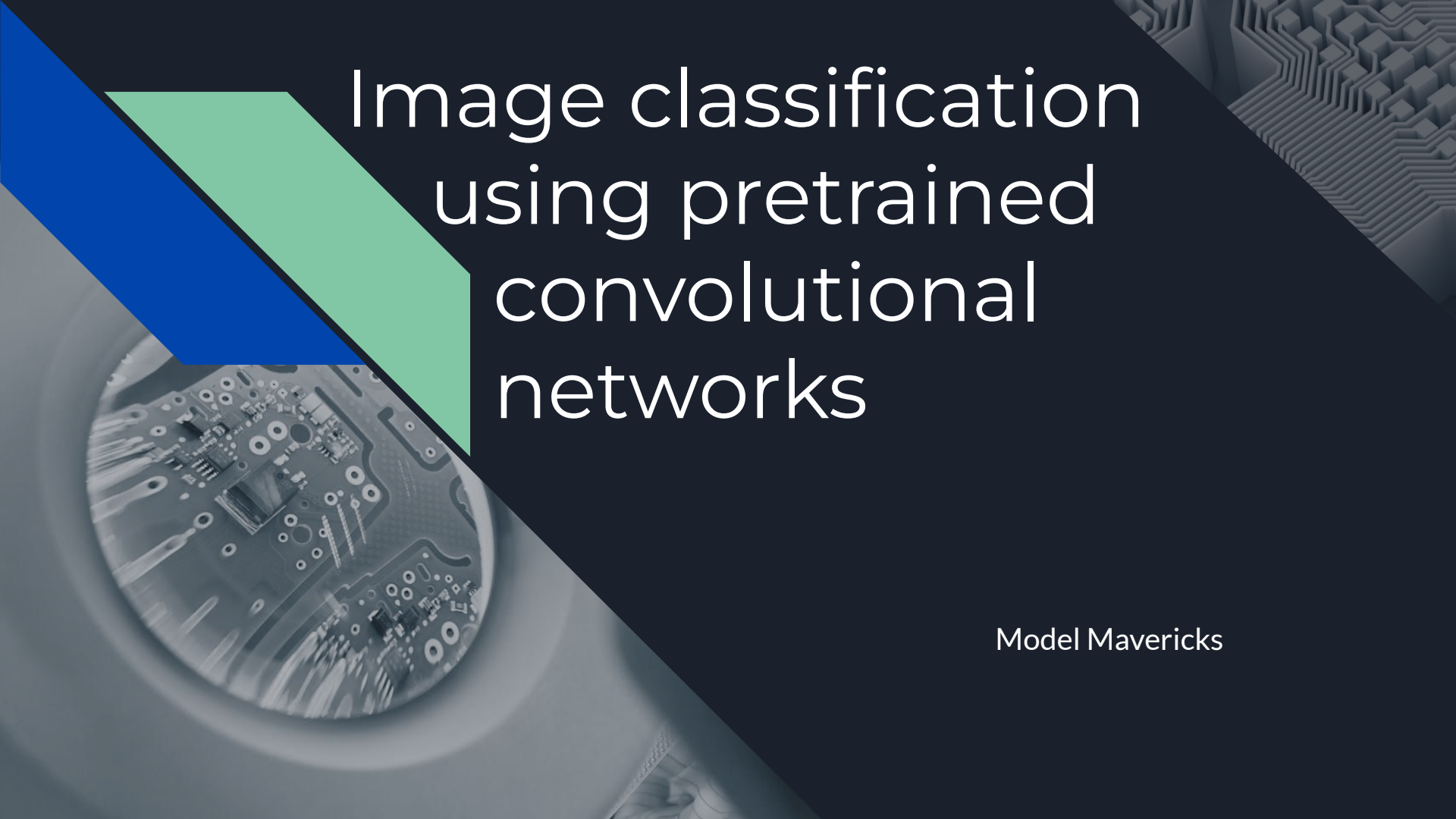
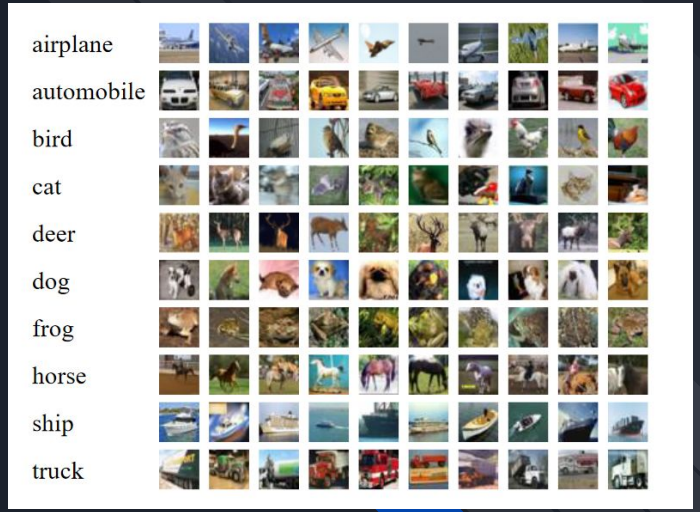
Image classification using pretrained convolutional networks



Model Mavericks

Introduction

- Project Goal: Compare advantages of pretrained neural networks versus randomly initialized ones for image classification.
- Key Focus: Highlight differences between pretrained and randomly initialized models
  - Convergence speed
  - Accuracy
  - Overall effectiveness



Previous Solutions

- CIFAR10 is widely used as a benchmark
- Many new papers, models try to use it to demonstrate capabilities (if applicable)
- Current SOTA uses Vision Transformer technology (600M+ params)

System Design

Containerization

- Yupiter Notebook
- Docker Compose

Data Processing

- Pytorch Tensor
- Augmentation

CNN from scratch

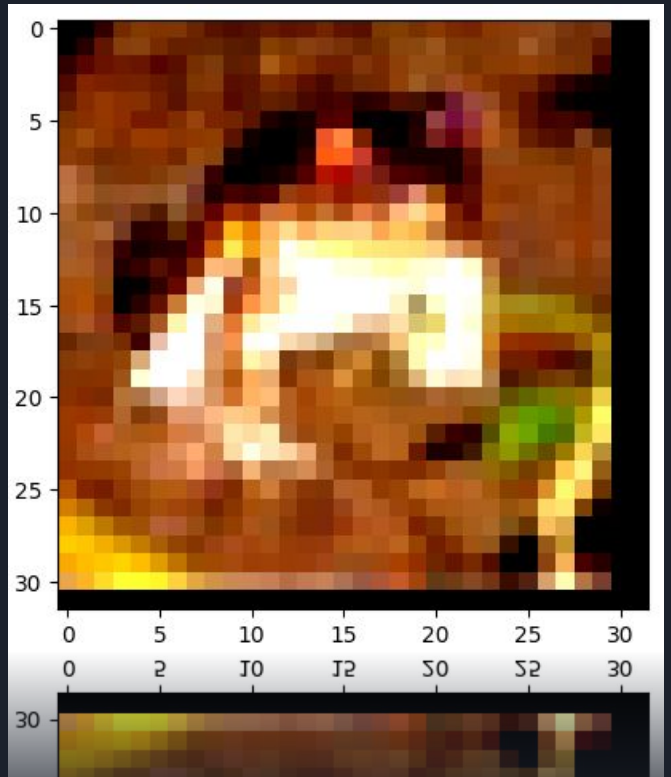
- 1.2 million parameters
- MaxPooling

ResNet18

- 11 million parameters
- Timm

About the Data - The CIFAR-10 Database

- 60,000 32x32 color images, 10 classes - 6,000 images per class
- Split into 50,000 training images and 10,000 test images
- Augmentations:
  - Random crop with padding
  - Random horizontal flip
  - Conversion to PyTorch tensor
  - Normalization



Training

- Manual hyperparameter-optimization
  - mainly LR, by factors of 10
- CrossEntropyLoss (for classification)
- Adam optimizer
- Utilized cuda

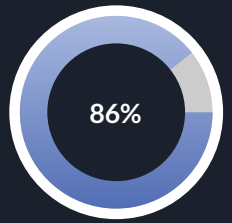
Results

ResNet18

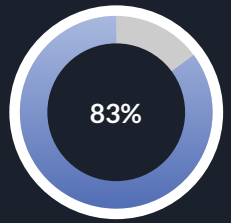
- Fine-tuning
- Pre-existing Knowledge
- Faster accuracy convergence

Our Model

- Smaller size
- Randomly initiated weights
- Slower convergence
- Lower overall accuracy



ResNet18
Test Accuracy



Our Model

Summary

- Results
  - Models performed as expected
  - Hands-on Experience with ML Tools
  - Insights regarding ML DevOps

- Challenges
  - Containerisation
  - Version Control