



REL Team

PROJECT 1: DEEP LEARNING-IMAGE
CLASSIFICATION WITH CNN

MEMBERS:

Alrumaysaa Alghamdi

Elaf Alkhalidi

Layla Alsulaimani

Contents



Project Introduction

Dataset

Preprocesses

DenseNet Overview

CNN inspired by DenseNet

Training Process

Training Results

DenseNet Transfer Learning

TF Results

Overall Results

Demo



Project Introduction

Project Objective:

Build and optimize an image classification model for CIFAR-10.

Approach:

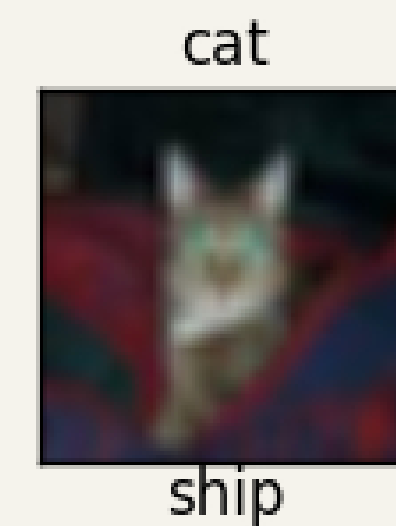
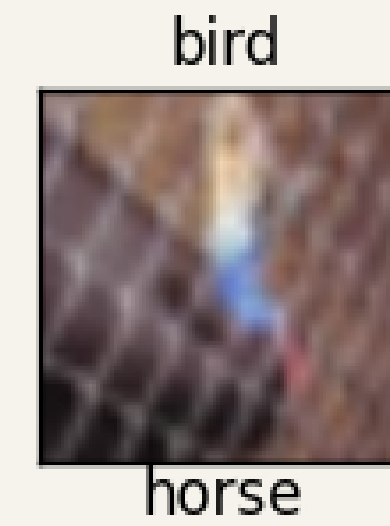
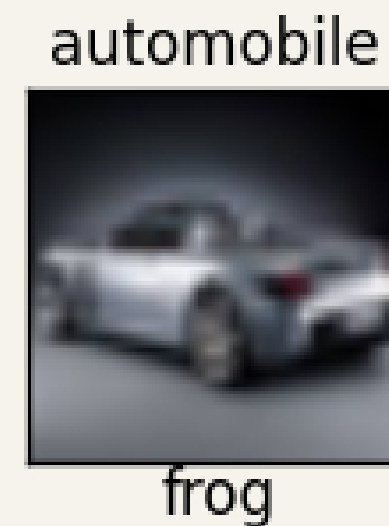
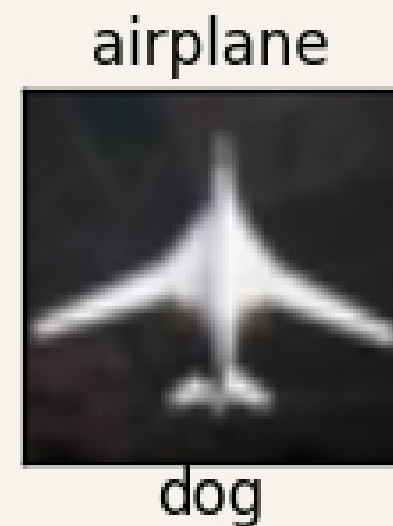
- Develop a custom CNN inspired by DenseNet, train from scratch.
- Perform transfer learning on the DenseNet model.
- Compare between models performance.

Dataset:



CIFAR-10 (Canadian Institute For Advanced Research)

- 60,000 colored images
- 10 classes
- 32x32 resolution



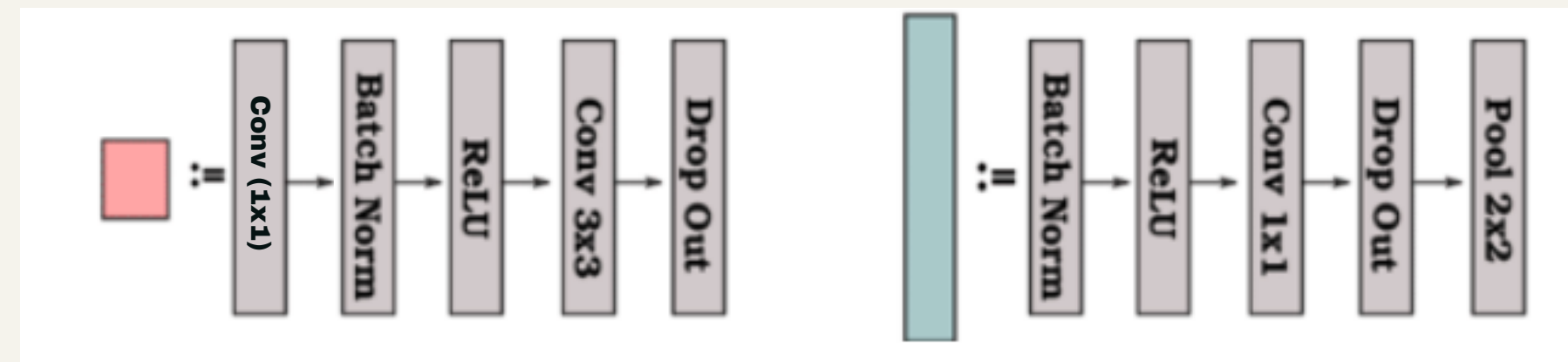
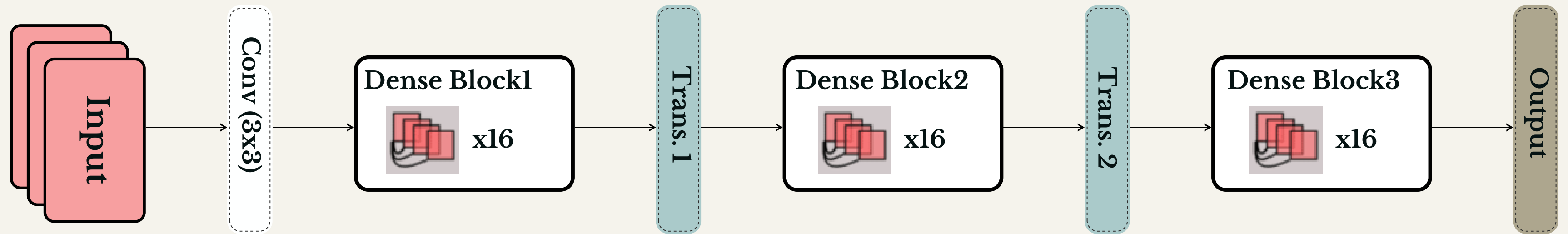
Preprocessing steps:



- 1 Load and split the data into train and test sets
- 2 **Normalization:**
Normalized pixel values to the range [0, 1] by dividing by 255.
- 3 **Label Encoding:**
Converted integer labels to one-hot encoded vectors using `to_categorical()`
- 4 **Data Augmentation:**
 - Applied transformations using `ImageDataGenerator` to increase dataset diversity:
 - Rotation Range: 15 degrees.
 - Width Shift: 10% of the image width.
 - Height Shift: 10% of the image height.
 - Horizontal Flip: Randomly flip images horizontally.
 - Fill Mode: 'nearest' to fill empty pixels after transformations.



Overview: DenseNet



Training Process



Densnet Hyperparameters:

- growth_rate = 12
- no. Layers/Dense Blocks = [16]*3
- compression= 0.5
- weight_decay= 1e-4
- dropout_rate= 0.3

Training Hyperparameters:

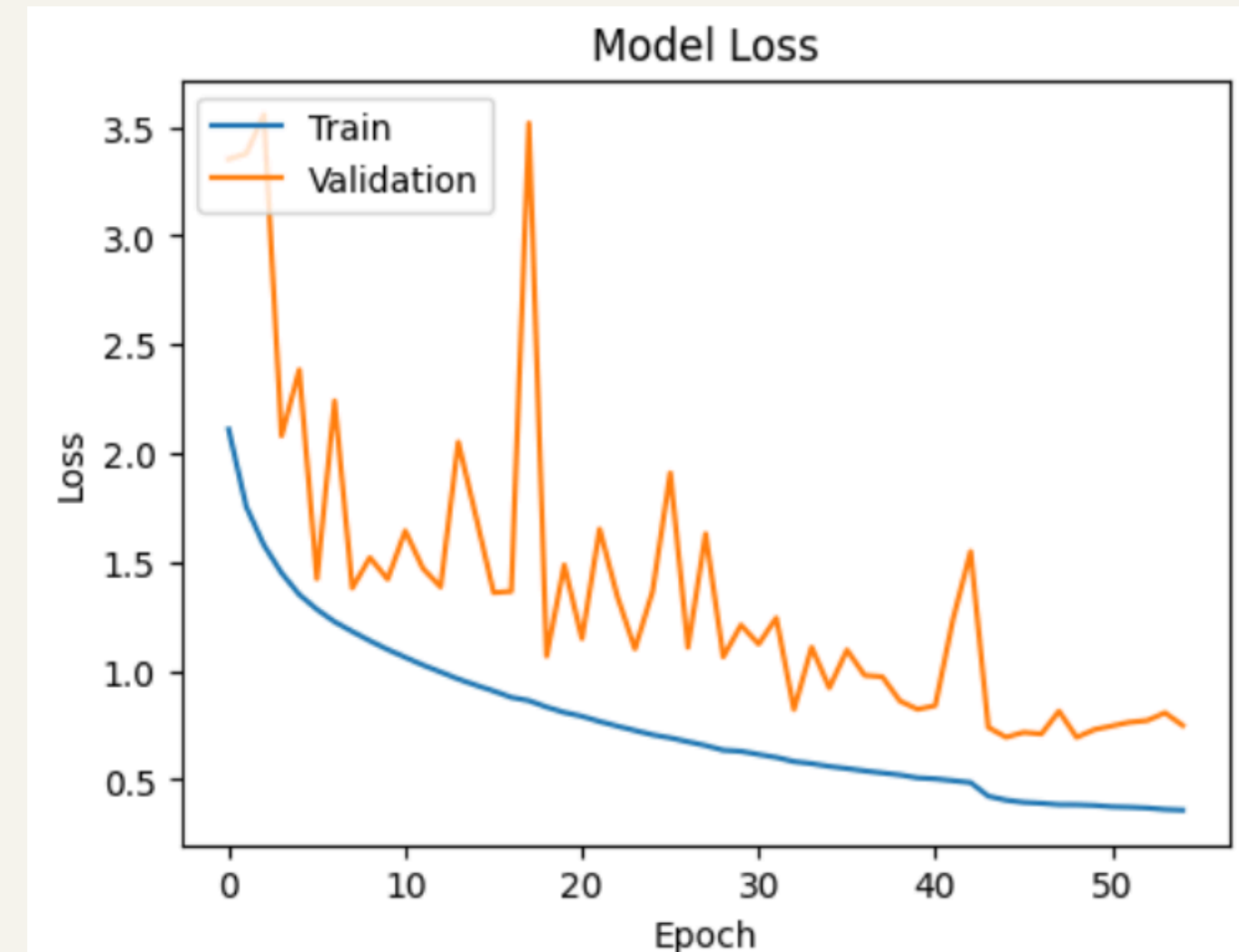
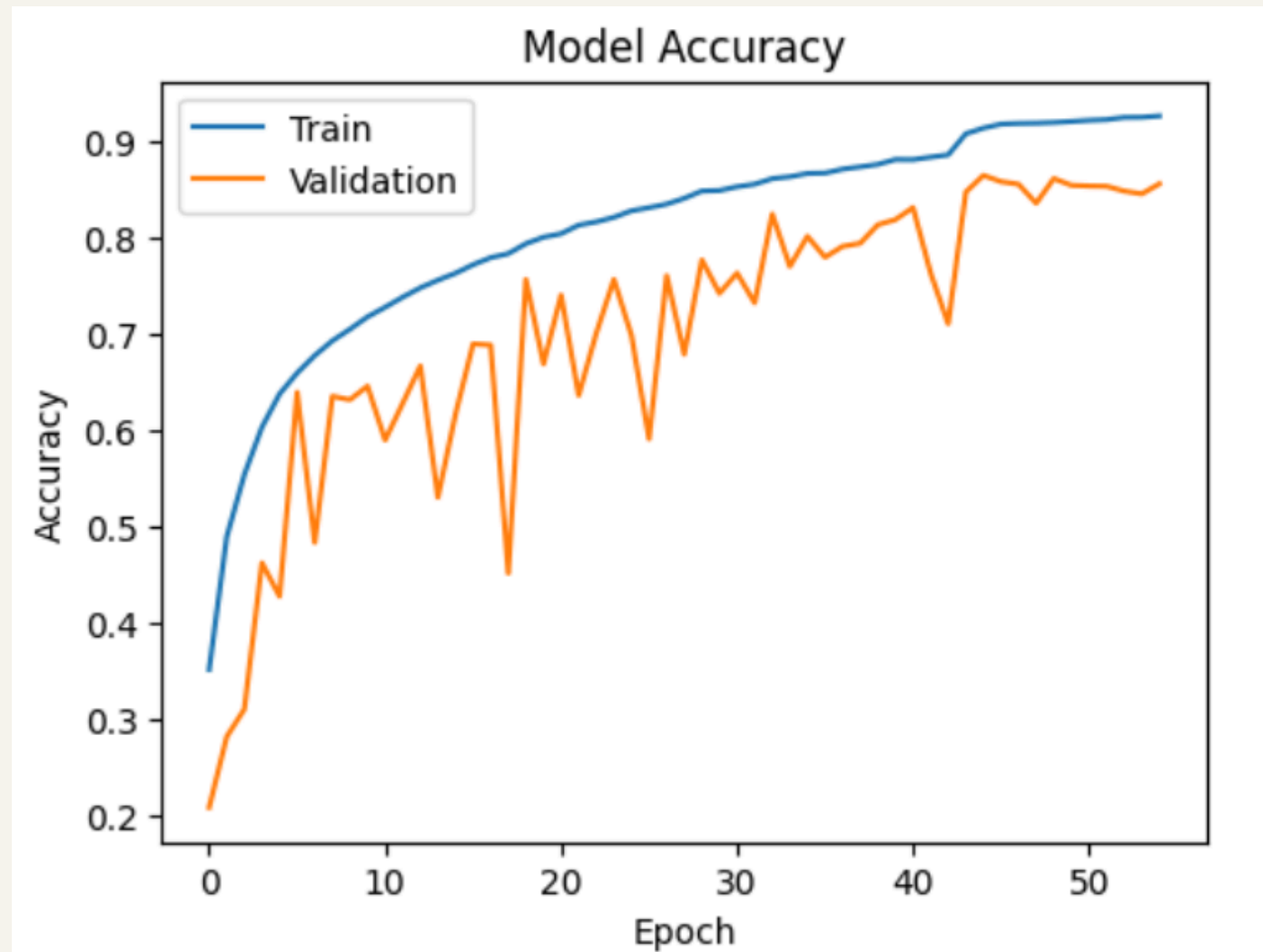
- batch_size = 64
- epochs = 100
- callbacks:
 - Stopping Early
 - Checkpoint
 - LR reducer
- Optimizer: SGD (LR=0.1)

DensNet Result



Accuracy: 86.09%

Loss: 69.44%



DenseNet -121 Transfer Learning



Layers:

- Upscaling: Resized Input img to (160x160)
- Base Model: Pre-trained DenseNet121 (without top layers).
- Custom Top Layers:
 - Flatten layer.
 - Batch Normalization.
 - Dense layers with ReLU activation.
 - Dropout layers.
 - Softmax output layer

Hyperparameters:

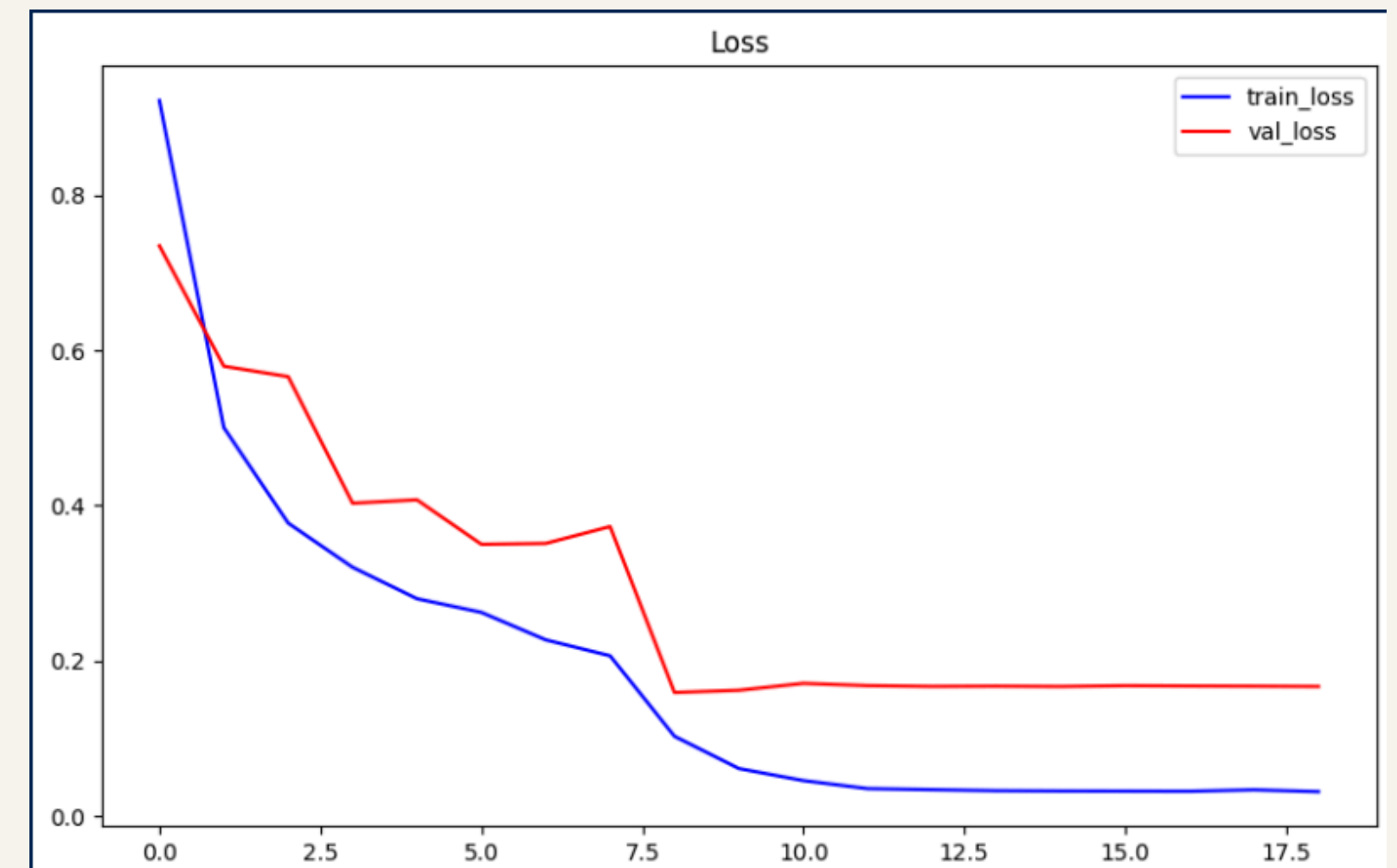
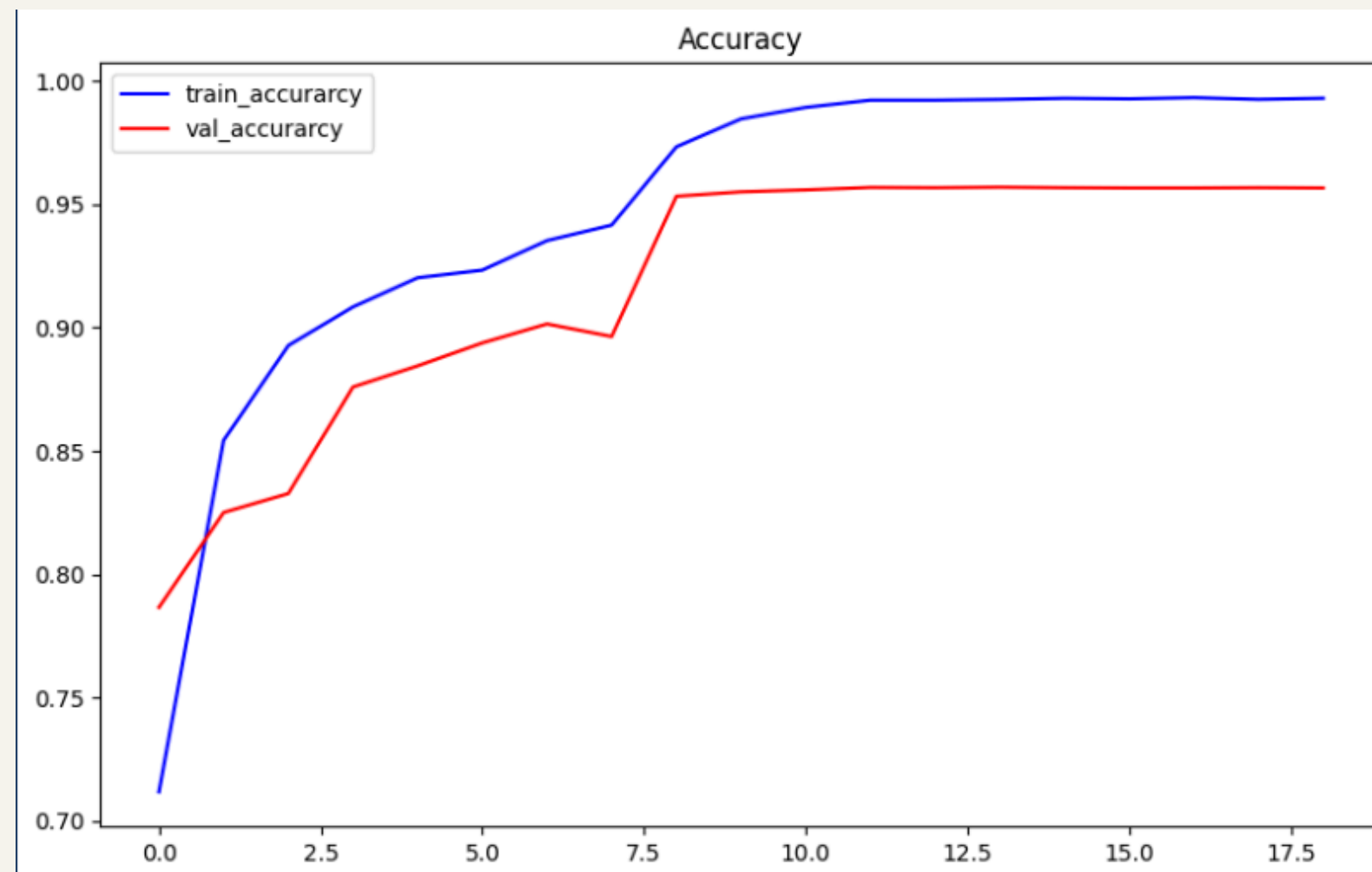
- batch_size = 128
- epochs = 20
- callbacks:
 - Stopping Early
 - Checkpoint
 - LR reducer
- Optimizer: Adam

DenseNet-121 TL Result



Accuracy: 95.9 %

Loss: 15.53%





Overall Result

Transfer Learning with DenseNet121:

Achieved 95.9% validation accuracy on the CIFAR-10 dataset

Custom Model (Inspired by DenseNet):

Achieved lower accuracy compared to transfer learning which was 86.09%

Challenges:

- CIFAR-10 images are small (32x32), requiring upscaling for DenseNet121.
- Balancing fine-tuning and overfitting due to the limited dataset size.

REL TEAM



Thank you for listening

ANY QUESTIONS?



Project Repo:

<https://github.com/Alrumaysaa-Alghamdi/cifar10-cnn-inspired-by-densenet-project>