**TechPart Vision: Final Project Report**

**Index:**

1. Introduction

1.1. Project overviews

1.2. Objectives

2. Project Initialization and Planning Phase

2.1. Define Problem Statement

2.2. Project Proposal (Proposed Solution)

2.3. Initial Project Planning

3. Data Collection and Preprocessing Phase

3.1. Data Collection Plan and Raw Data Sources Identified

3.2. Data Quality Report

3.3. Data Preprocessing

4. Model Development Phase

4.1. Model Selection Report

4.2. Initial Model Training Code, Model Validation and Evaluation Report

5. Model Optimization and Tuning Phase

5.1. Tuning Documentation

5.2. Final Model Selection Justification

6. Results

6.1. Output Screenshots

7. Advantages & Disadvantages

8. Conclusion

9. Future Scope

10. Appendix

10.1. Source Code

10.2. GitHub & Project Demo Link

**1. Introduction**

**1.1 Project Overviews**

The project "TechPart Vision” aims to develop a robust image classification model to accurately classify and identify various personal computer(PC) parts. Utilizing the EfficientNet architecture and transfer learning techniques, this project leverages pre-trained models to enhance classification accuracy and efficiency.

**1.2 Objectives**

* To build a high-accuracy image classification model for PC parts.
* To aid in activities such as invemtory management, online shopping and technical support.

**2. Project Initialization and Planning Phase**

**2.1 Define Problem Statement**

The primary challenge addressed by this project is the automatic and accurate identification of personal computer parts from images, which is essential for inventory management, online marketplaces, and technical support.

**2.2 Project Proposal (Proposed Solution)**

The proposed solution involves using EfficientNet for transfer learning to develop a classification model that can identify different PC parts. The model will be trained on a labelled dataset of PC part images and will undergo several optimizations to achieve high accuracy.

**2.3 Initial Project Planning**

* Identify and collect a diverse dataset of PC part images.
* Data preprocessing to bring it up to standard
* Choosing the appropriate model for high accuracy and efficiency
* Evaluation/validation followed by deployment

**3. Data Collection and Preprocessing Phase**

**3.1 Data Collection Plan and Raw Data Sources Identified**

Dataset was collected from Kaggle, whose raw data was scraped from google images.

**3.2 Data Quality Report**

* Dataset had issues such as missing values and inconsistent image sizes.
* Solutions such as resizing images, and reducing the number to match the least represented group were implemented.

**3.3 Data Preprocessing**

* Resizing and normalizing images for consistent input to the model.
* Augmenting the dataset with transformations such as rotations, flips, and brightness adjustments to enhance model robustness.
* Splitting the dataset into training, validation, and test sets.

**4. Model Development Phase**

**4.1 Model Selection Report**

**Initial Model: Custom CNN**

**Structure:**

* Layers: Convolutional, MaxPooling, Dropout, Flatten, Dense
* Activation: ReLU (convolutional and dense), Softmax (final dense)
* Optimizer: Adam
* Loss Function: Categorical cross-entropy
* Training: 10 epochs, batch size 32, verbose 1

**Performance:**

* Result: Underfitting; low accuracy

**Transition to EfficientNetV2B1**

**Rationale:**

* Selected for its efficiency and performance, suitable for deployment with limited resources
* Utilizes pre-trained ImageNet weights for transfer learning
* Enhanced accuracy with fewer training samples

**Custom Layers Added:**

* GlobalAveragePooling for dimension reduction
* BatchNormalization for training stability
* Dense layer with ReLU and regularization to prevent overfitting
* Dropout layer for further overfitting reduction
* Softmax activation in output layer for classification into 14 categories

**Performance:**

* Significant improvement in training and validation accuracy and loss
* Outperformed VGG19

**Comparison with VGG19**

**Structure:**

* VGG19 without top classification layer (include\_top=False)
* Added: GlobalAveragePooling2D, two Dense layers (1024 units with ReLU, final layer with softmax)

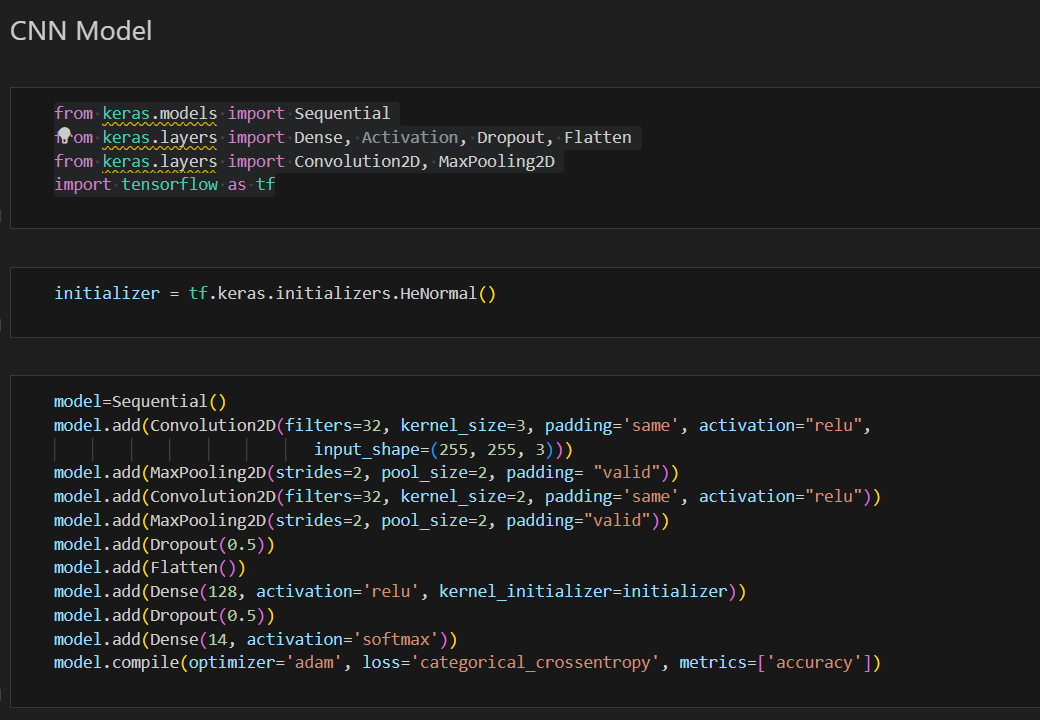
**Performance:**

* Initial improvements but validation accuracy plateaued, indicating overfitting

**Transition Justification:**

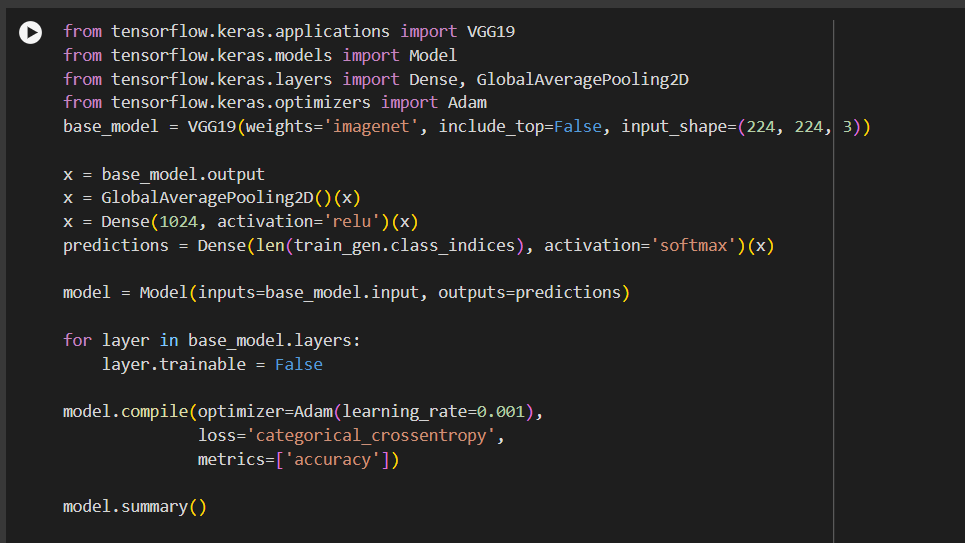
* EfficientNetV2B1 showed better training and validation accuracy, justifying its selection over VGG19.

4.2**. Initial Model Training Code, Model Validation and Evaluation Report**

****

**A screenshot of a computer program

Description automatically generated**

****

**5. Model Optimization and Tuning Phase**

**5.1 Tuning Documentation**

**CNN Sequential model:**

Convolutional Layers: Number of Filters: 32; Kernel Size: 3 (first layer), 2 (second layer); Activation Function: ReLU; Padding: Same

Pooling Layers: Pooling Size: 2

Regularization: Dropout Rate: 0.5

Dense Layers: Units: 128; Kernel Initializer: HeNormal

Output Layer: Units: 14

Training Parameters: Batch Size: 32; Number of Epochs: 10

**EfficientnetV2B1:**

Base Model

* Base Model: EfficientNetV2B1 (pretrained on ImageNet, used as a feature extractor)

Pooling

* Pooling: Max (global max pooling applied to the output of the base model)

Batch Normalization

* Axis: -1
* Momentum: 0.99
* Epsilon: 0.001

Dense Layer

* Units: 256

Regularization

* Kernel Regularization: l2(0.016)
* Activity Regularization: l1(0.006)
* Bias Regularization: l1(0.006)

Activation

* Activation Function: ReLU

Dropout

* Rate: 0.4

Output Layer

* Units: 14 (softmax activation for multi-class classification)

Optimizer

* Optimizer: Adamax
* Learning Rate: 0.001

Callbacks

* ReduceLROnPlateau:
  + Monitor: "val\_loss"
  + Factor: 0.4
  + Patience: 2
  + Min LR: 0.0
* EarlyStopping:
  + Monitor: "val\_loss"
  + Patience: 2
  + Restore Best Weights: True

Training Parameters

* Number of Epochs: 5

**VGG19:**

Base Model

* Base Model: VGG19 (pretrained on ImageNet, used as a feature extractor)
* Input Shape: (224, 224, 3)

Pooling

* Pooling: Global Average Pooling

Dense Layers

* Units: 1024
* Activation Function: ReLU

Output Layer

* Units: len(train\_gen.class\_indices)
* Activation Function: Softmax

Optimizer

* Optimizer: Adam
* Learning Rate: 0.001

Loss Function

* Loss Function: Categorical Crossentropy

Metrics

* Metrics: Accuracy

Freezing Layers

* Freezing Layers: Initially freeze all base model layers

**5.2 Final Model Selection Justification**

The comparison between VGG19 and EfficientNetV2B1 shows that EfficientNetV2B1 outperforms VGG19 in several key areas. EfficientNetV2B1 achieves higher training accuracy more quickly and maintains a relatively stable validation accuracy around 0.75, while VGG19's validation accuracy fluctuates and remains generally lower. Both models exhibit a significant decrease in training loss, but EfficientNetV2B1 starts with a higher value and decreases more sharply initially. Additionally, EfficientNetV2B1's validation loss shows a more consistent decrease, whereas VGG19's validation loss fluctuates after the initial decrease. Overall, EfficientNetV2B1 demonstrates better performance with higher and more stable validation accuracy and a consistent decrease in validation loss, indicating quicker and more reliable improvements compared to VGG19.

**6. Results**

This project was able to classify various PC parts such as keyboards, monitors, cables, RAM, HDD etc, and identify them with significant accuracy.

**A collage of computer mouses

Description automatically generated**

From website:



**7. Advantages & Disadvantages**

**Advantages**

* Significant accuracy in classifying PC parts.
* Efficient and fast training with EfficientNet.
* Transfer learning reduces the need for a large dataset.

**Disadvantages**

* Requires significant computational resources.
* Performance is dependent on the limited dataset.

**8. Conclusion**

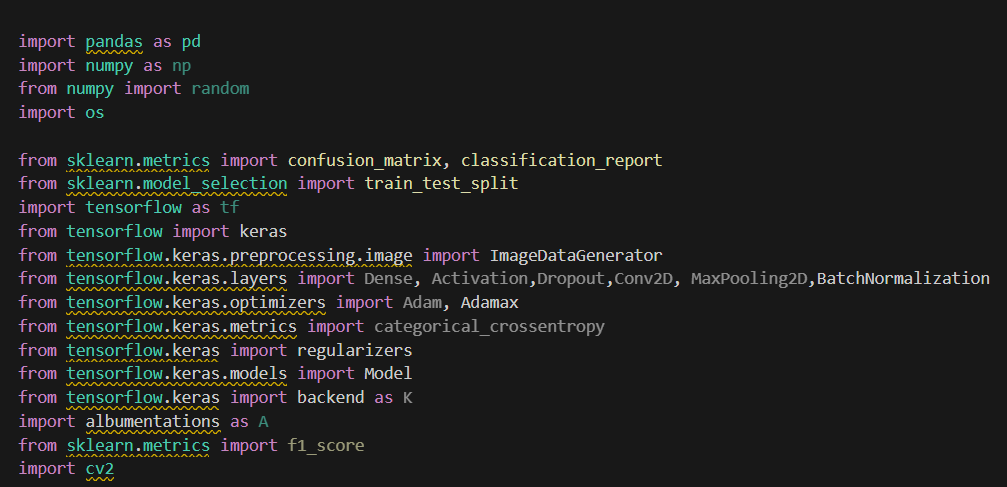
The project successfully implemented EfficientNet for classifying PC parts with high accuracy. The model's performance demonstrated the effectiveness of transfer learning for image classification tasks.

**9. Future Scope**

* Extending the model to classify more types of PC parts.
* Integrating the model into real-time inventory or e-commerce systems.
* Exploring other advanced architectures and techniques for further improvement.

**10. Appendix**

**4.1 Source Code**

****

**A black rectangular object with white lines

Description automatically generated**

**A black rectangular object with a black stripe

Description automatically generated**

**A screen shot of a computer

Description automatically generated**

**A black screen with white text

Description automatically generated**

**A black rectangular object with a black stripe

Description automatically generated**

**A screen shot of a computer

Description automatically generated**

**A black rectangular object with a black stripe

Description automatically generated**

**A black rectangular object with a black line

Description automatically generated**

**A computer screen shot of code

Description automatically generated**

****

**A screenshot of a computer program

Description automatically generated**

**A screenshot of a computer program

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer program

Description automatically generated**

**A screenshot of a computer program

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer program

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A black background with orange and white text

Description automatically generated**

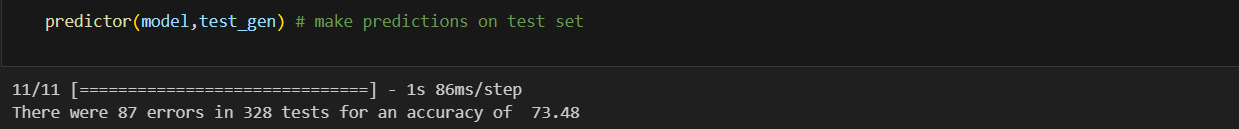
**A comparison of a graph

Description automatically generated with medium confidence**

****

**A screen shot of a computer code

Description automatically generated**

**A grid of blue squares

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screen shot of a computer

Description automatically generated**

**A screen shot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A computer code on a black background

Description automatically generated**

**A collage of computer mouses

Description automatically generated**

**4.2 Github and Project demo link**

**Project Demo link:**

[**https://vitapacin-my.sharepoint.com/:v:/g/personal/kowshik\_22bce9556\_vitapstudent\_ac\_in/Ef3AmwMXiQZNu2fK-iNm1c4BcC77mJ2tRjCON\_-WhX1GKw?nav=eyJyZWZlcnJhbEluZm8iOnsicmVmZXJyYWxBcHAiOiJTdHJlYW1XZWJBcHAiLCJyZWZlcnJhbFZpZXciOiJTaGFyZURpYWxvZy1MaW5rIiwicmVmZXJyYWxBcHBQbGF0Zm9ybSI6IldlYiIsInJlZmVycmFsTW9kZSI6InZpZXcifX0%3D&e=vevLJx**](https://vitapacin-my.sharepoint.com/:v:/g/personal/kowshik_22bce9556_vitapstudent_ac_in/Ef3AmwMXiQZNu2fK-iNm1c4BcC77mJ2tRjCON_-WhX1GKw?nav=eyJyZWZlcnJhbEluZm8iOnsicmVmZXJyYWxBcHAiOiJTdHJlYW1XZWJBcHAiLCJyZWZlcnJhbFZpZXciOiJTaGFyZURpYWxvZy1MaW5rIiwicmVmZXJyYWxBcHBQbGF0Zm9ybSI6IldlYiIsInJlZmVycmFsTW9kZSI6InZpZXcifX0%3D&e=vevLJx)

**Github:**

[**https://github.com/tsadityaa/TechPart-Vision**](https://github.com/tsadityaa/TechPart-Vision)

[**https://github.com/gauthamt1008/TechPart-Vision**](https://github.com/gauthamt1008/TechPart-Vision)

**https://github.com/kowshikdontu/** **TechPart-Vision**