Final Project Report: AI Model Comparison

Course: Introduction to AI
Project Title: Comparative Analysis of Three AI Models
Submitted by: Natalie Carrero, Cole Gilliam, Julianne Maghirang
Date: December 12, 2024

1. Introduction and Objective

Objective:
The goal is to build three AI Models capable of classifying the make and model of a car based on an uploaded image. Each model should accurately identify the car's brand and model name from a large dataset using image classification.

Problem Statement:
Due to the large variety of car makes and models, these AI models are designed to automate car make and model identification from images which can be useful in various industries, such as the automotive industry, insurance, and surveillance.

Overview of AI Models Chosen:

Model Number	Model Name	Purpose
1	Convolutional Neural Network (CNN)	Serves as the baseline model for image classification using raw pixel data.
2	Autoencoders + Logistic Regression	Reduces dimensionality and uses a lightweight classifier for car detection.
3	XGBoost	Uses extracted features for highly efficient and accurate car classification.
4	ResNet-50	Leverages deep residual networks to improve performance in image classification.

2. Justification of Model selection

Model Name	Reason for Selection
Convolutional Neural Network (CNN)	CNNs are highly effective at automatically extracting spatial features from image data, making them ideal for image classification tasks.
Autoencoders + Logistic Regression	Autoencoders reduce the dimensionality of the image data, enabling more efficient classification, and logistic regression offers a simple yet effective classifier for distinguishing features.
XGBoost	XGBoost provides high accuracy and efficiency for classification tasks when used with features extracted from a CNN, offering robust performance and regularization to prevent overfitting.
ResNet-50	ResNet is capable of high performance in image classification by mitigating issues such as vanishing gradient and enables the use of deeper architectures.

3. Model Descriptions

Model Overview:

Model #	**Model Name**	**Architecture Details**	**Key Features**
1	Convolutional Neural Network (CNN)	Contains multiple convolutional layers and pooling layers for classification. Extracts spatial features for images	- High accuracy for image classification - Efficient feature extraction - Spatial hierarchies in data
2	Autoencoders + Logistic Regression	Encodes and decodes layers, where the encoded output is passed to a Logistic Regression classifier.	- Effective dimensionality reduction - Combines unsupervised (autoencoder) and supervised (regression) learning
3	XGBoost	A gradient-boosted decision tree algorithm optimized for speed and performance. Uses boosting to create an ensemble of weak learners.	- High performance for tabular data - Handles missing values well - Regularization techniques to avoid overfitting
4	ResNet-50	A deep residual network with skip connections to alleviate the vanishing	- Capable of training very deep networks.

		gradient problem, allowing for the training of very deep networks.	- High accuracy in image recognition tasks

4. Dataset Description

The AI Model Comparison Project utilizes a dataset pulled from kaggle that is a subset of the CompCars Dataset, a large variety of car images containing various backgrounds, angles, and differing car makes and models.

Dataset Attribute	Image Classification of .jpg files
Name	Stanford Cars Data (subset of CompCars dataset)
Source	Download available: https://www.kaggle.com/datasets/jessicali9530/stanford-cars-dataset With further information from: https://mmlab.ie.cuhk.edu.hk/datasets/comp_cars/
Size	16,185 images
Class Distribution	196 classes of cars with 8,144 training images and 8,041 testing images, where each class has approximately a 50-50 split.
Preprocessing Steps	1. Image Resizing 2. Normalize images 3. Label Encoding

Dataset Justification:

The Stanford Cars Data was chosen due to its large coverage of car makes, models, and conditions. Additionally, it already provided the preprocessing steps of excluding all images that only included car parts (not an entire car) and two file folders for training and testing. The provided images contained various backgrounds and angles to enhance the accuracy and robustness of the model. Finally the dataset also provided a labels file that includes the classification for each image.

5. Experimental Setup

Experimental Design:

Metric
Accuracy
Precision
Recall
F1 Score

Parameter Settings:

Model Name	Batch	Epochs	Learning Rate	Layers
CNN	32	10	0.001	3 (Conv)
Logistic Regression + Autoencoders	N/A	N/A	Implicit (solver = saga)	500 (PCA)
XGBoost	N/A	N/A	N/A	N/A
ResNet50	32	25	0.001	50

Environment Details:

Component	**Specification**
Operating System	Google Colab Runtime
Software Version	Python, TensorFlow, Keras
Hardware	Python 3 Google Compute Engine
Link to the code base	https://github.com/jmaghirang/AI-Car-Classifier

6. Results and Analysis

Performance Metrics:

Model Name	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
CNN	76.38%	78.81%	76.38%	76.15%
Logistic	1.2%	1.2%	0.91%	0.75%
XGBoost	4%	5%	7%	5%
ResNet	86%	88%	87%	87%

Comparative Analysis:
Logistic regression: This model performed suboptimally compared to the other models. This is in part to the nature of the goal, car classification. Logistic regression struggles to capture the complex patterns in the images.

XGBoost: It struggled with image classification because it relies on extracted features rather than processing raw pixel data like CNNs and ResNet. Without highly detailed feature extraction, its accuracy was limited. While it's good at tabular data and regularization to prevent overfitting, its performance was restricted due to the less effective feature representation.

ResNet: This model outperformed all because it uses skip connections to avoid common issues like losing important information during training. These connections allow the model to build deepers layers and learn complex patterns in the car images. As a result, it could classify the cars even with different backgrounds or angles, making it the most effective for this dataset.

CNN: Outperformed the simpler models, showing strong performance in classifying car images. CNNs are effective at extracting the spatial hierarchies from images, but it was not as accurate as ResNet for this particular task.

Error Analysis:

Logistic regression: There are many factors that hinder the model created. Firstly the train data and the test data were limited due to high computing time. The data for this project reached over 32k pictures including the test and the train. Training the data over this stretch was impractical in the time left so it was narrowed down. Another factor could be extracted features not being meaningful enough or detailed enough.

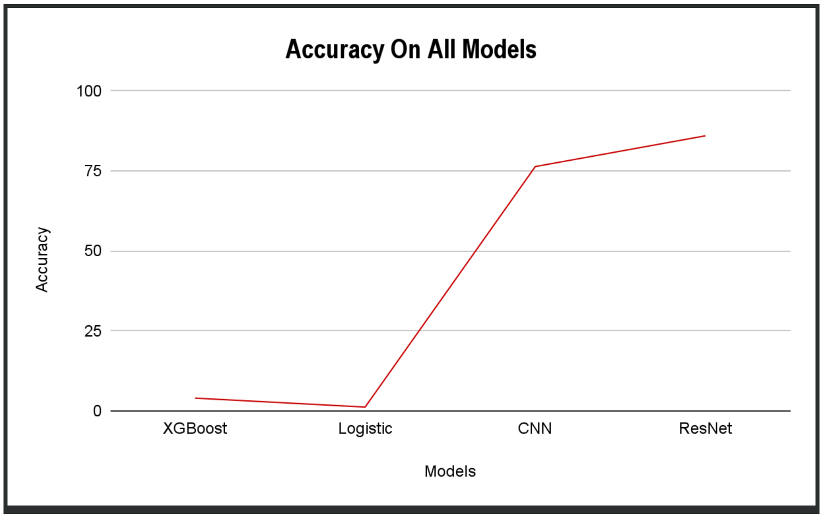
XGBoost: As mentioned, its performance was limited due to feature extraction. It depends on meaningful features, which are challenging to obtain from image data. Also, limited hyperparameter tuning due to time constraints may have caused underperformance. With more feature engineering, it could have improved.

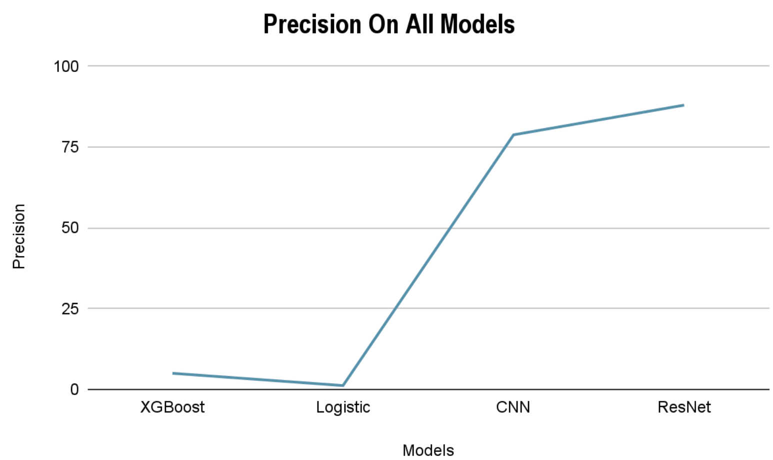
ResNet: The biggest challenge was its need for high computing power and memory. The limited size of the dataset also increased the chance of overfitting, meaning the model could perform too well on the training data but struggle with new images. Since training took a long time, there wasn't enough time to adjust settings like learning rate and batch size, which could have improved the results.

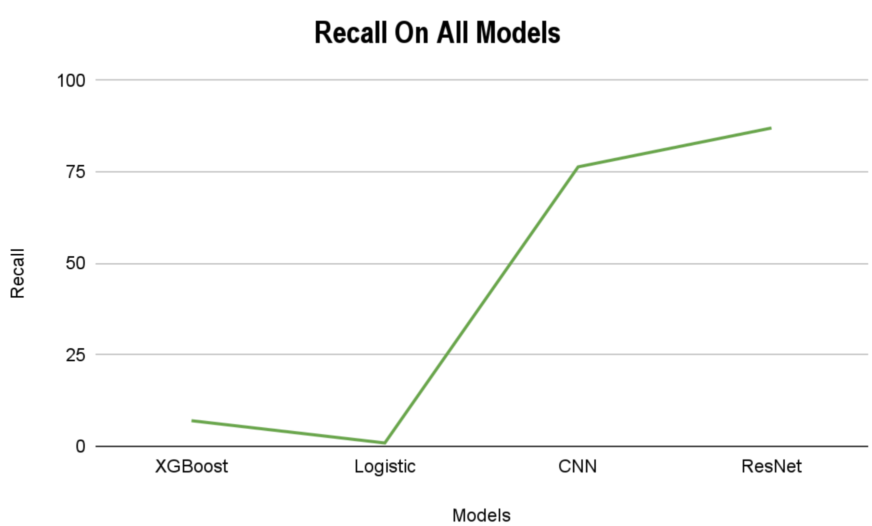
CNN: Training for this particular model requires tuning various hyperparameters (learning rate, batch size, epochs, etc.) which may not be fully optimized due to time and how expensive computations can be.

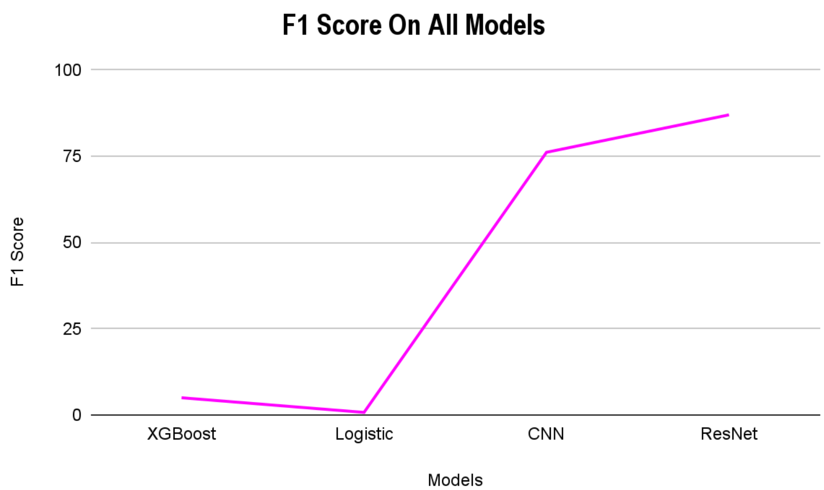
Plots:

Metrics Across All Models

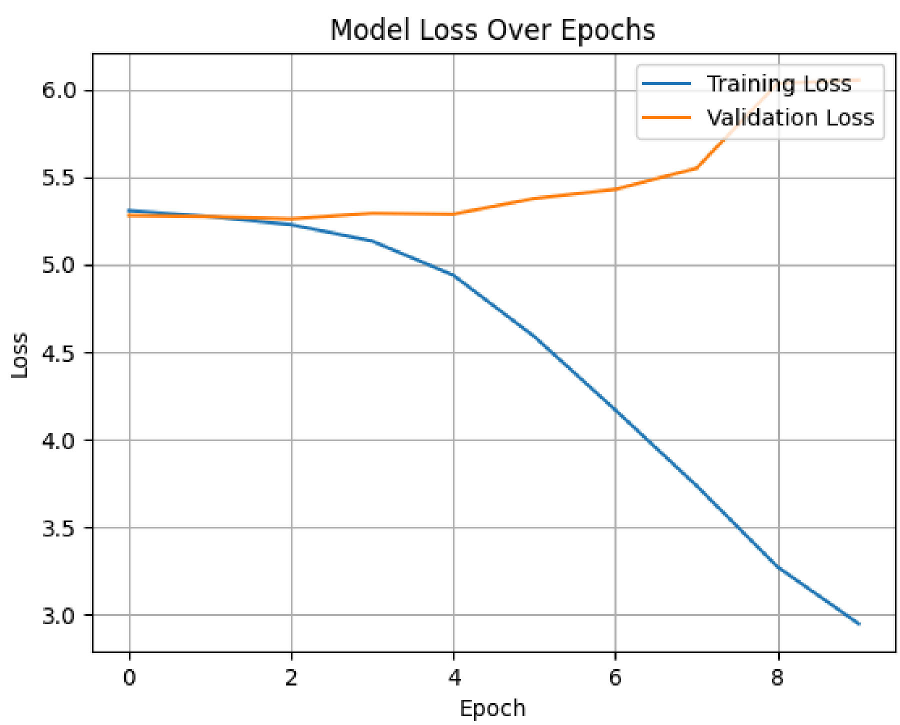

Accuracy On All Models
100
75
50
25
0
Accuracy
XGBoost
Logistic
CNN
ResNet
Models


Precision On All Models
100
75
50
25
0
Precision
XGBoost
Logistic
CNN
ResNet
Models


Recall On All Models
100
75
50
25
0
Recall
XGBoost
Logistic
CNN
ResNet
Models



For the CNN model:
Demonstration of overfitting, as the model performs well on the training data but not on validation data



7. Discussion and Insights

Interpretation of Results:
In general, CNN and ResNet outperformed Logistic Regression and XGBoost by a landslide in

terms of classifying car makes and models. This was to be expected since deep learning models are more well-suited for complex tasks such as image classification. ResNet in particular performed the best due to its deep architecture and skip connections which prevent the vanishing gradient problem.

Limitations:
1. Computationally Expensive - Particularly with the more complex models, the models are highly computationally expensive, especially when dealing with image data. Limitation on powerful hardware drastically slowed down the training process.

2. Dataset Size - The total dataset size was 16,185 images which for deep learning models (CNN and ResNet) is relatively small. This could lead to overfitting since these models rely on large datasets to learn complex features and patterns.

3. Time Constraints - Due to time constraints, training a singular version of a model can take up most of the day to the entire day. Overall it did not allow for extensive hyperparameter tuning. Due to this, the models may not have achieved their best possible performance.

Future Directions:
1. Expand on the dataset to improve model performance. An ideal dataset would be about 50k-100k images. This will also allow for more representation of different car makes and models.

2. With more time, optimizing hyperparameters would likely improve results. With that, introducing different data configurations, such as cross-validation, grid search or random search to improve hyperparameters and general performance.

3. Experimentation with other architectures that may yield higher accuracy, such as EfficientNet or Vision Transformers (ViTs).

8. Conclusion

Project demonstrated that deep learning models are far more effective for car classification tasks than traditional machine learning models. ResNet, in particular, achieved the best overall performance, showcasing its ability to handle complex image data.

Strengths:

ResNet: Proven to work well on large-scale image datasets and capable of effectively classifying complex images by using deep networks without suffering from degradation.

CNN: Flexible in terms of complexity, in other terms, can be designed to be simple or deep depending on the task at hand. The hierarchical structure of CNNs allows for incremental improvements and different refinement settings based on the task.

XGBoost: Benefits from gradient boosting and supports parallelization which is great for faster training on large datasets.

Logistic Regression: Computationally efficient and can be regularized to prevent overfitting which would be ideal for smaller datasets.

Weaknesses:

ResNet: Requires substantial memory resources due to its large model size and can be held back by memory or computational restrictions. Certain networks can lead to very long training times.

CNN: Requires large amounts of labeled training data to avoid overfitting leading it to be less effective on small datasets. Altering parameters can lead to significant increase to training time and difficulty to optimize.

XGBoost: Prone to overfitting, particularly with small datasets or hyperparameters that are not correctly optimized. Requires effective feature extraction that could otherwise limit its performance on tasks.

Logistic Regression: Struggles to model non-linear relationships directly, which limits its performance on complex datasets. The model's performance can also be negatively impacted by irrelevant or redundant features.