Comparative Analysis between End-to-End Models & Multi-Staged Models for Accident Detection using CCTV frames

1st Aditya Aggarwal
CSAI Department
Indraprasth Institute of Information Technology
Delhi, India
aditya22028@iiitd.ac.in

2nd Mehul Pahuja
CSAI Department
Indraprasth Institute of Information Technology
Delhi, India
mehul22295@iiitd.ac.in

***Abstract*— Road traffic accidents result in a significant number of fatalities annually, with delayed Emergency Medical Services (EMS) response times exacerbating the issue. Our project aims to solve this problem with the best possible accuracy. For this, various models have to trained and tested. This brings us to the main objective of the paper, to provide a comparative analysis between End to End Learning Models and Multi-Staged Learning Models. For this, we will explore both Machine Learning and Deep Learning techniques.**

Index Terms—End to End Model, Multi-Staged Model, Accident Detection, Machine Learning, Deep Learning, CNN

I. Problem Statement And Motivation

Our problem statement is to provide a comparison between E2E models and Multi-Staged Models for image classification tasks. For this we have chosen the task of accident detection using the CCTV footage frames. This problem is very crucial in order to improve road safety and response time of medical services. The main motivation of this paper is determine for our use case, and in general for image classification tasks, which approach is best, End to End or Multi-Staged. The ultimate result of this may even help traffic control department and Government to enhance road safety and reduce fatalities.

II. Literature Review

A. An Introduction to Variable and Feature Selection

With several features available for selection in a Multi-Staged model, it is important to know which features produce the desired results.

B. End-to-end learning every purpose ML method

The ultimate question is whether End-to-End learning is the ultimate replacement for the traditional Multi-Staged models. For an example, in the field of audio based models & Speech Recognition, Multi-Stage Pipeline is as follows: Audio (input) -> feature extraction -> phoneme detection -> word composition -> text transcript (output). Whereas an E2E model has a simple pipeline Audio (input) — — — (NN) — → transcript (output).

DataSet Details

For the task of accident detection, multiple datasets were available on Kaggle and other resources. The following datasets were considered:

I. Name: Road Accidents from CCTV Footages
Source: https://www.kaggle.com/datasets

II. Name: Accident Detection from CCTV Footage
Source: https://www.kaggle.com/datasets

III. Name: Annotated images taken from the video feed from traffic cameras
Source: https://open.canada.ca/

However, there were various issues with the provided datasets. Some of them had misclassification in true labels in the train set, while some of them had inadequate size of dataset. One major problem was that the datasets contained sequential frames which were arbitrarily distributed over the directory. To tackle all these issues, we manually tailored a custom dataset using relevant images and labels from all of these resources.

Our dataset consists of **36, 022** training images. Since this is a binary classification problem, we had two unique labels, 1 for accident and 0 for non-accidents. A validation set of size 2160 data-points was created which contained equal number of images from both labels.

TABLE I
Distribution of Datasets

Dataset	*Train*	*Val*	*Test*
Accident	10,469	1,080	649
Non-Accident	25,553	1,080	899
Total	36,022	2,160	1,548

After obtaining the dataset we preprocessed the dataset. We made the dimensions of each image uniform by resizing them to 240 by 240 pixels. We converted the images to Numpy arrays where each entry denoted the value of each pixel of the image. We have also normalized every image.

IV. Methodology

1. Data Preprocessing:

- Load accident and non-accident images from numpy files.
- Create labels for each dataset indicating accidents (1) and non-accidents (0).
- Balance the number of data points for each class by randomly selecting a subset of samples to make them equal.
- Flatten the input data for further processing.

2. Dimensionality Reduction using Incremental PCA:

- Apply Incremental PCA (IPCA) for dimensionality reduction.
- Fit IPCA on the flattened training data.
- Transform both the training and test data using the fitted IPCA model.
- Save the IPCA model and transformed data for future use.

3. Model Training and Evaluation:

- Apply different classification algorithms on the IPCA-transformed data:
    - Quadratic Discriminant Analysis (QDA)
    - AdaBoost Classification
    - Gradient Boosting
    - Random Forest
    - Deep Layer Neural Network (DNN)
- Train each model on the transformed training data and evaluate its performance on the transformed test data using accuracy as the evaluation metric.
- Display the accuracy of each model.

4. Visualizing Predictions:

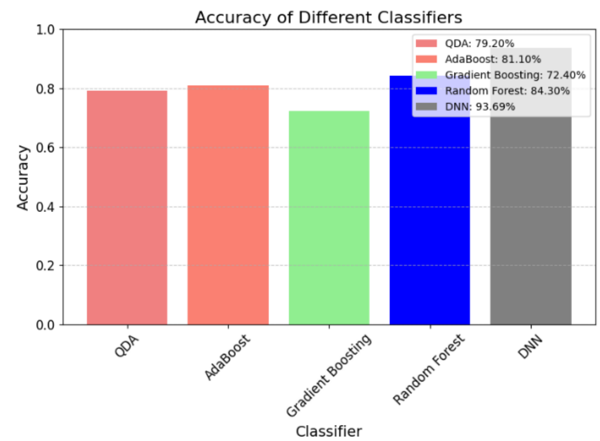
- Plot 5 random images of accidents and non-accidents from the test data along with their corresponding labels using our most accurate model (DNN).

5. Presentation:

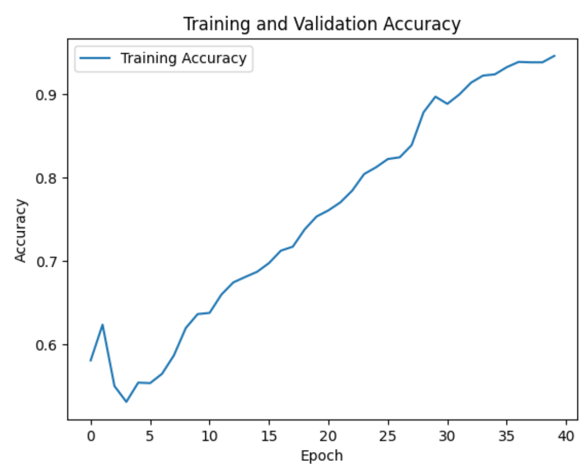
- Plot the accuracies of different classifiers to compare their performance visually.
- Specify the accuracy of each classifier alongside its name and use different colors for better differentiation.

V. Visualization

Accident Vs Non-Accident Plot



Different Model Accuracies



Accuracy vs Epoch

REFERENCES

[1] Deep Learning for Emotion Recognition on Small Datasets Using Trans-fer Learning Hong-Wei Ng, Viet Dung Nguyen, Vassilios Vonikakis, Stefan Winkler Advanced Digital Sciences Center (ADSC) Universityof Illinois at Urbana-Champaign, Singapore

[2] Article: "E2E: The Every-Purpose ML Method" https://towardsdatascience.com/e2e-the-every-purpose-ml-method-5d4f20dafee4