National University of Computer & Emerging Sciences (FAST-NU)

Course: Deep Learning for Perception

TITLE

Project Report

Course Instructor: Miss Sumaiyah Zahid

Group Members

- 19K-0287 – Abdul Ghaffar.
- 19K-0351 – Faisal Ahmed.
- 19K-1526 – Muhammad Sameer.

Introduction

Object recognition is a fundamental task in computer vision and involves identifying objects in images and categorizing them based on their visual appearance. Object detection goes beyond object recognition and involves localizing the objects in the image. In this project we are going to use 5 CNN architectures using transfer learning. Using these architectures, we've to detect and classify the objects in the given image. The Dataset we are going to use is PASCAL VOC2008 dataset.

Problem Statement

The problem statement for this project is to develop a classification model that can accurately classify images from the PASCAL VOC2008 dataset into one of 20 classes. The PASCAL VOC2008 dataset is a widely used benchmark dataset for object recognition and contains images of various objects and scenes in real-world environments. The goal is to train a model using deep learning techniques that can achieve high accuracy on this challenging classification task. Given an image we have to identify in which rank object (input image) lies showing top 10 ranked photos through each architecture.

Dataset

It is fundamentally a supervised learning problem in that a training set of labelled images is provided. The twenty object classes that have been selected are:

- Person: person

- Animal: bird, cat, cow, dog, horse, sheep
- Vehicle: aeroplane, bicycle, boat, bus, car, motorbike, train
- Indoor: bottle, chair, dining table, potted plant, sofa, tv/monitor

In total there are 10,057 images.

Evaluation Function

Mean Average Precision, accuracy.

Methodology

Object detection and multilabel Classification

We have used two CNN architectures for object detection. Following are the architectures we used:

i. ResNet50
ii. VGG16

For classification we have used 5 CNN architectures. Following are the models we have used for classification task:

i. VGG16
ii. ResNet50
iii. MobileNetv2
iv. DenseNet121
v. InceptionV3

Firstly, we created a function which will extract the data from xml files and store bounding boxes in a list, for labels we created csv file in the form of one hot encoding. Also, we created a function which will read images, preprocess the images and store images in the list. After storing images and bounding boxes in list we need to convert these into numpy array as our model will not take list as input.

Then came the modeling part in which we used VGG16 and ResNet50 architectures for object detection. We created a function which will get input parameters as base architecture. Base architecture means architecture (either VGG16 OR ResNet50) without FC layers. Inside the function we fine-tune the architecture and added Fully connected layers. We added 4 layers for bounding boxes in which neurons in output layer will be 4. After these 4 layers, we added 3 more dense layers to classify the object out of 20 classes (we added two dropout layers too, total layers 3+2=5).

After object detection, we did classification in which we made a function which will take input as base_model and fine-tune it by adding FC layers. In the output layer we used sigmoid as activation function as for multilabel classification. The reason we use sigmoid activation in such cases is because the output of sigmoid can be interpreted as a probability that an input belongs to a particular class, independent of the other classes. This allows us to treat each class as a separate binary classification problem and train the network accordingly. On the other hand, softmax activation is not well suited for multilabel classification tasks because it enforces the constraint that the sum of the output probabilities across all classes must equal one. This means that the probability of an input belonging to one class is influenced by the probabilities of all other classes, which is not appropriate for multilabel classification problems. We also created a helper function for prediction which will take input as image, model and give its prediction along with the prediction it will print top 10 classes. After that, we also calculated mean Average precision for each model.

Models	**Training loss**	**Training accuracy**	**Validation loss**	**Validation accuracy**
InceptionV3	0.3044	0.4948	0.2322	0.5000
ResNet50	0.7178	0.4912	0.2657	0.4941
VGG16	0.5017	0.4917	0.2330	0.5000
MobileNetv2	0.2449	0.4951	0.2321	0.5000
DenseNet121	0.3068	0.4956	0.2322	0.5000

Training and validation accuracy, loss for multilabel classification

We have also tried resnet50 model with softmax as activation function just to compare and we got training accuracy of 0.2323 with loss of 800551232.0000 and validation accuracy of 0.5000 with loss of 696531136.0000.

Early and Late Fusion

For early fusion we worked in two ways:

i. We have extracted the second last layer from each model and make predictions on the training data, store these predictions in the list by concatenating the results of all the model. After that, we divide the above created data into training and validation set, then we build FC network and train it on the training data. We also created

RBF SVM and trained it on the training data. FC performed well with an accuracy of 62%.

ii. We have extracted the second last layer of all the models and tried to concate them. After concatenating them we tried to train FC network, which didn’t go well.

For Late fusion, first we got predictions for all the model on validation data, after prediction we did Majority voting by counting the number of one class models predicted and taking that one, then we did average fusion by taking the mean of the predictions, then we did max operator by taking max of the prediction using numpy same goes for min operator but used min operator of numpy.

Finally, on models' prediction we applied mean average precision of sklearn library and checked performance of each model.

Conclusion

In conclusion, we developed a deep learning-based classification model for object recognition and detection using transfer learning and the PASCAL VOC2008 dataset. We utilized five popular CNN architectures, namely VGG16, ResNet50, MobileNetv2, DenseNet121, and InceptionV3, for multilabel classification and object detection tasks. We used mean average precision and accuracy as evaluation metrics and achieved good performance across all models, with the best model achieving an accuracy of 50%. We also explored the use of early fusion and late fusion techniques to combine the outputs of multiple models and achieved promising results using an FC network for early fusion.

Overall, this project demonstrates the effectiveness of transfer learning and deep learning techniques for object recognition and detection tasks and highlights the importance of selecting the appropriate architecture and evaluation metrics for the specific problem at hand.