Video Label Classification
CS 688

Term Project

--------- Team ---------

Neeraj Fernandes (nferna10@gmu.edu)
Prakhar Dogra (pdogra@gmu.edu)
Yogen Chaudhari (ychaudha@gmu.edu)

DATASET

- Dataset chosen for our video label classification is the Youtube-8M Dataset
- This dataset contains over 7 million YouTube videos.
- Video Files are stored in tfrecord format files
- Each tfrecord file has approximately 1200 videos
- There are total of 4716 labels (each video with multiple labels)
- Each video can have up to 10 labels

Our Approaches

We have implemented two approaches for video label classification

Approach A: Classifying one frame at a time with a CNN (Single Frame Model)

Approach B: Long term recurrent convolutional networks (LRCN)

Approach A

- For this approach we have implemented a 2D CNN architecture.
- For the 2D CNN we have ignored temporal features of videos and attempted to classify each video by looking at a single frame
- Also, as part of demo and experiments we have used only 720,000 videos during training (approximately 16% of the available training data).
- Optimizers: “rmsprop” and “adam”
- Global Average Precision on the testset is 0.5899401951067841

Approach A : High-Level Architecture Diagram

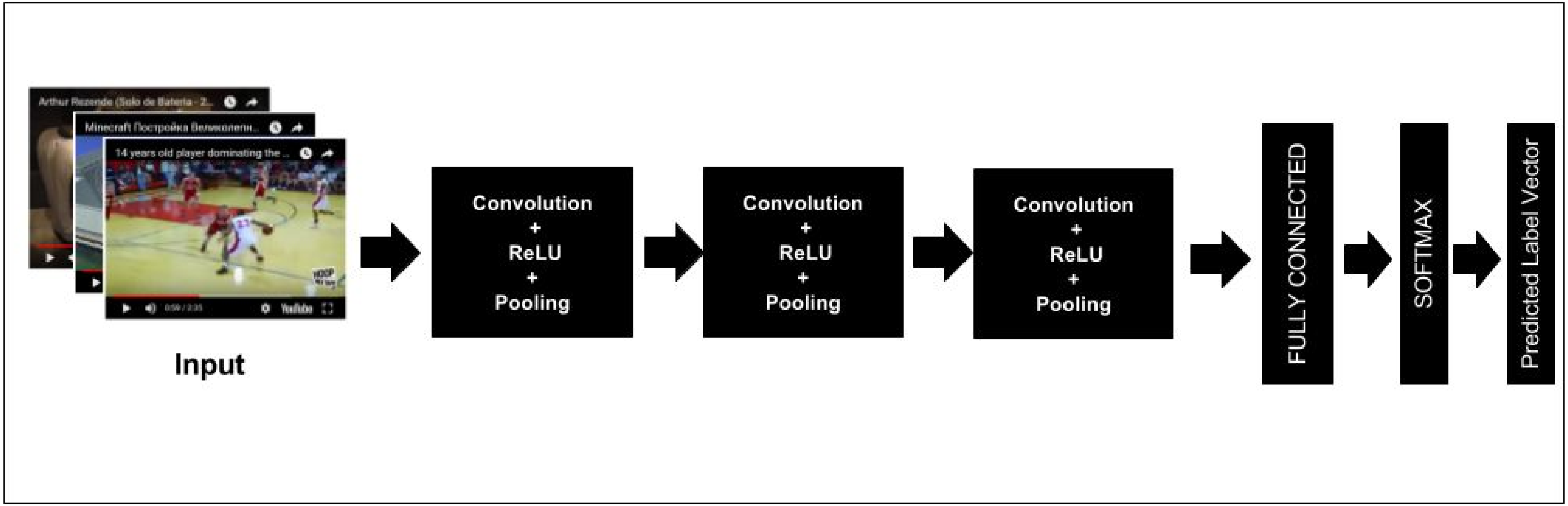

Figure: High Level Architecture Diagram Approach A

Approach A : Low-Level Architecture Diagram

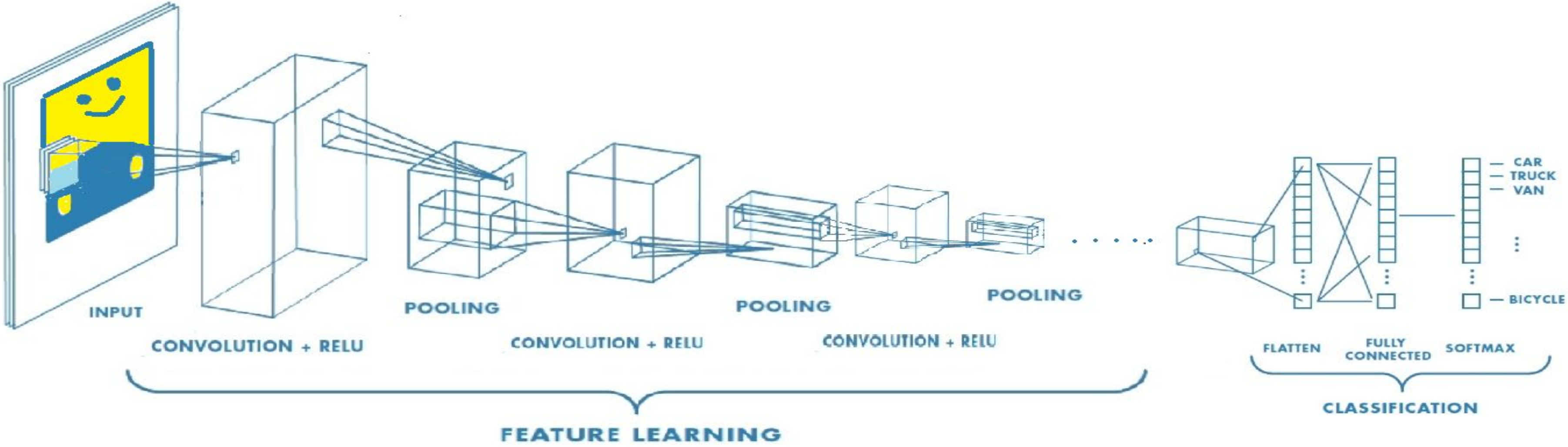

Figure: Feature Learning Using CNN

Model for Approach A:

```
#Model
model = Sequential()
model.add(Conv2D(32,(3,3), input_shape = (32, 32, 1)))
model.add(Activation("relu"))
model.add(MaxPooling2D(pool_size = (2, 2)))

model.add(Conv2D(32,(3,3)))
model.add(Activation("relu"))
model.add(MaxPooling2D(pool_size = (2, 2)))

model.add(Conv2D(64,(3,3)))
model.add(Activation("relu"))
model.add(MaxPooling2D(pool_size = (2, 2)))

#Dropout to prevent overfitting
model.add(Flatten()) #flatten feature map into 1 dimension
model.add(Dense(64))
model.add(Activation("relu"))
model.add(Dropout(0.2))

#final layer to predict probabilities
model.add(Dense(4716))
model.add(Activation("softmax"))
```

Approach B

- For this approach we have implemented Long term recurrent convolutional networks (LRCN)
- We have used Convolutional Neural Network to extract features from each frame
- Then we have passed the sequence of extracted features (frames) to a separate RNN (Recurrent Neural Network)

Approach B : High-Level Architecture Diagram

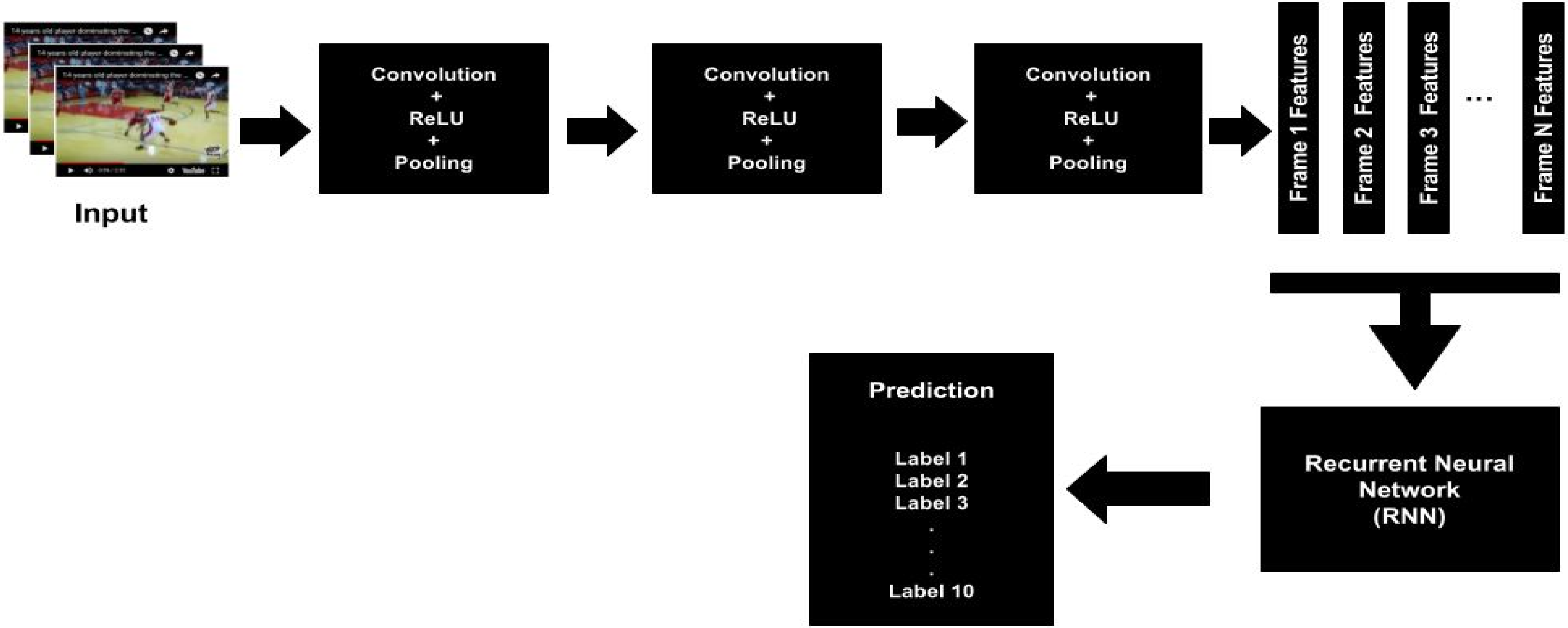

Figure: High Level Architecture Diagram Approach B

Model for Approach B:

```
#Model
model = Sequential()

model.add(TimeDistributed(Conv2D(32, (7,7), strides=(1, 1), activation='relu',
                                  padding='same'), input_shape=(120, 32, 32, 1)))
model.add(TimeDistributed(Conv2D(32, (3,3), kernel_initializer="he_normal", activation='relu')))
model.add(TimeDistributed(MaxPooling2D((2, 2), strides=(2, 2))))


model.add(TimeDistributed(Conv2D(64, (3,3), padding='same', activation='relu')))
model.add(TimeDistributed(Conv2D(64, (3,3), padding='same', activation='relu')))
model.add(TimeDistributed(MaxPooling2D((2, 2), strides=(2, 2))))

model.add(TimeDistributed(Conv2D(128, (3,3), padding='same', activation='relu')))
model.add(TimeDistributed(Conv2D(128, (3,3), padding='same', activation='relu')))
model.add(TimeDistributed(MaxPooling2D((2, 2), strides=(2, 2))))


model.add(TimeDistributed(Conv2D(256, (3,3), padding='same', activation='relu')))
model.add(TimeDistributed(Conv2D(256, (3,3), padding='same', activation='relu')))
model.add(TimeDistributed(MaxPooling2D((2, 2), strides=(1, 1))))

model.add(TimeDistributed(Conv2D(512, (3,3), padding='same', activation='relu')))
model.add(TimeDistributed(Conv2D(512, (3,3), padding='same', activation='relu')))
model.add(TimeDistributed(MaxPooling2D((2, 2), strides=(1, 1))))

model.add(TimeDistributed(Flatten())) #flatten feature map into 1 dimension

model.add(Dropout(0.2)) #Dropout to prevent overfitting
model.add(LSTM(256, return_sequences=False, dropout=0.2))
model.add(Dense(4716, activation='softmax'))
```

Example 1


```
Actual Labels are:
Game
Athlete
Basketball moves
Point guard
School
nan
Highlight film
Basketball
Slam dunk


Predicted Labels are:
Game
Basketball
Basketball moves
Highlight film
Association football
Slam dunk
Athlete
Wrestling
Stadium


Number of labels that were found common : 6 out of 9
```

Example 2

	Actual Labels are: Cymbal Snare drum Drum Drummer Drum kit Predicted Labels are: Cymbal Snare drum Drummer Drum kit Drum Number of labels that were found common : 5 out of 5

Example 3

Actual Labels are:
Minecraft
Castle

Predicted Labels are:
Game
Video game

Number of labels that were found common : 0 out of 2

Testing & Evaluation

We have used Holdout Method for validation and testing for the following reasons:

- We have an enormous amount of data to train on. So we aren't losing out on the number of available examples.
- It takes huge amount of time for a single iteration of training and testing so using k-fold cross validation isn't a good option.

We have used Global Average Precision score as evaluation metric

Global Average Precision (GAP)

The evaluation takes the predicted labels that have the highest k confidence scores for each video, to compute the Average Precision across all of the predictions and all the videos.

If a video has N predictions sorted by its confidence score, then the Global Average Precision is computed as:

$$GAP = \sum_{i=1}^{N} p(i)\Delta r(i)$$

where N is the number of final predictions (if there are 20 predictions for each video, then N = 20 * number of Videos), p(i) is the precision, and r(i) is the recall.

Source: https://research.google.com/youtube8m/workshop2017/c16.pdf

RESULTS

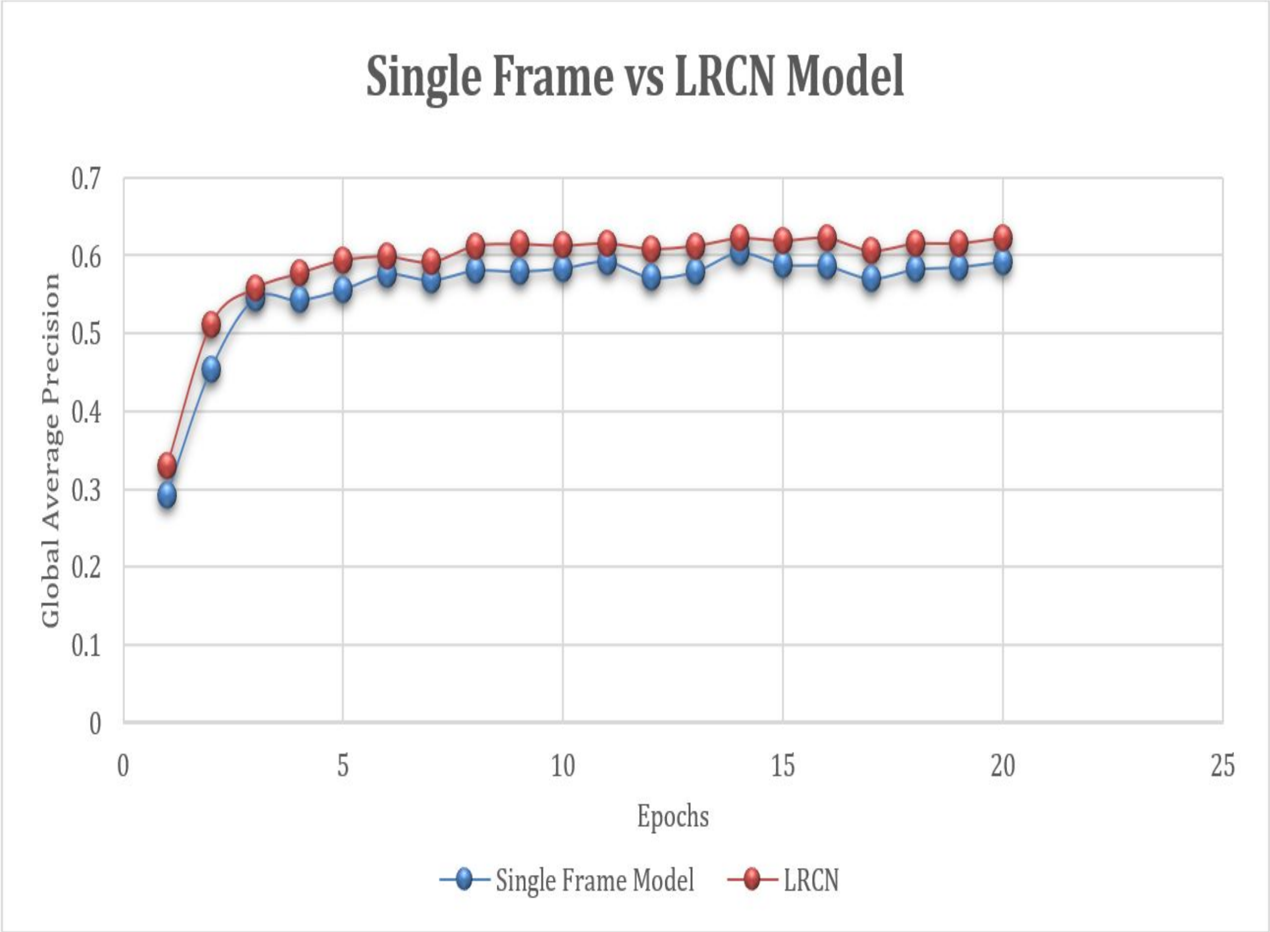

Figure: Comparing results of both approaches

Demo

Thank You !