NEWS IMAGE CLASSIFICATION

Ayush Shirsat, Youn Soo Park, Sunitha Priyadarshini Sampath

April. 8, 2019

Contents

1 Introduction

According to Information Handling Services (IHS), there were 245 million professionally installed video surveillance cameras active globally in 2014. Hypothetically, 10 cameras were watched by a person, we would need 24.5 million people. While 10 cameras were watched by a person, attacks of terrorist or violent protests rise over the years. The lack of resources of the security forces has made it extremely difficult to detect violent crowd flows and video cameras need constant human supervision which is costly and prone to human error.

This issue can be solved by using a deep learning model which detects events with protest. Since videos are sequence of images, a deep learning model can be implemented on a set of images to simplify the process. For the issue, we have studied Resnet and VGG for comparison.

2 Background Survey

2.1 Resnet

For our study on resnet model, we found a paper on "Protest Activity Detection and Perceived Violence Estimation from Social Media Images" by Donghyeon Won , Zachary C. Steinert-Threlkeld and Jungseock Joo. The paper addresses the problem of identifying violence in social media images using a Resnet Convolutional neural network

The paper describes the development of a visual model that recognizes protesters, describes their actions (by recognizing features such as fire, sign etc that are addressed as visual attributes) and the level of perceived violence. Their model and approach differs from existing models that use Natural Language Processing to study protests from images. NLP models use hashtags and links that describes images to predict violence. These approaches however, might not be efficient since the tags used depend on individuals that can often be misleading. Further information such as gender, race and emotions cannot be determined from tags or links.

2.1.1 Dataset

Their dataset consists of tweet images from 2013-2017 and other online resources. It consists of 40,764 images (11,659 protest images and hard negatives) in total with various annotations of visual attributes and sentiments. Upon analysis we found that their dataset and annotations are the same as the one presented to us for the problem.

Dataset Statistics

# of images: 40,764

# of protest images: 11,659

Protest & Visual Attributes

Fields	Protest	Sign	Photo	Fire	Police	Children	Group>20	Group>100	Flag	Night	Shouting
# of Images	11,659	9,669	428	667	792	347	8,510	2,939	970	987	548
Positive Rate	0.286	0.829	0.037	0.057	0.068	0.030	0.730	0.252	0.083	0.085	0.047

Fig 2.1.1 Image classification across protest and visual attributes

Attribute	Description
Sign	A protester holding a visual sign (on paper, panel, or wood).
Photo	A protester holding a sign containing a photograph of a person (politicians or celebrities)
Fire	There is fire or smoke in the scene.
Law enf.	Police or troops are present in the scene.
Children	Children are in the scene.
Group 20	There are roughly more than 20 people in the scene.
Group 100	There are roughly more than 100 people in the scene.
Flag	There are flags in the scene
Night	It is at night.
Shout	One or more people shouting.

Fig 2.1.2 List of Visual Attributes

2.1.2 Model and Approach

They developed two separate models to recognize protest activities in images. First, they trained a CNN which takes a full image as input and outputs a series of prediction scores including the binary image category (i.e., protest or non-protest) , visual attributes, and perceived violence and image sentiment . The model architecture is based on a 50-layer ResNet [19], which consists of 50 convolutional layers with batch normalization and ReLU layers.

Layer	Output size	Building blocks			
conv1	112 × 112	7 × 7, 64, stride 2			
conv2	56 × 56	3 × 3 max pool, stride 2			
		$\begin{bmatrix} 1\times1, & 64 \\ 3\times3, & 64 \\ 1\times1, & 256 \end{bmatrix} \times 3$			
conv3	28 × 28	$\begin{bmatrix} 1\times1, & 128 \\ 3\times3, & 128 \\ 1\times1, & 512 \end{bmatrix} \times 4$			
conv4	14 × 14	$\begin{bmatrix} 1\times1, & 256 \\ 3\times3, & 256 \\ 1\times1, & 1024 \end{bmatrix} \times 6$			
conv5	7 × 7	$\begin{bmatrix} 1\times1, & 512 \\ 3\times3, & 512 \\ 1\times1, & 2048 \end{bmatrix} \times 3$			
pooling	2048	average pooling			
classification	17	1-d fc (protest)	1-d fc (vio-lence)	4-d fc (senti-ment)	10-d fc (visual attribute)

Fig 2.1.3 Architecture of the model

The model uses binary cross-entropy loss to train the binary variables (protest and visual attributes) and mean squared error to train violence and sentiment dimensions.

Second, they trained another CNN that captures various facial attributes from images. Their analysis includes facial attributes and they argue that its is especially important because social scientists have theorized about the role of emotions in leading to and sustaining protests. They have used OpenFace for the face model, which was developed for face recognition. They used the CelebA facial attribute dataset to train the attribute model. For each image, they used dlib's face detection and alignment 4 and cropped the internal facial region to feed into the facial CNN model.

2.1.3 Result:

Protest Scene and Attributes Classification:

Fields	Protest	Sign	Photo	Fire	Law	Children
Pos. rate	.286	.829	.036	.057	.067	0.030
AUC	.969	.919	.738	.984	.921	.813
Fields	·	Group > 20	Group> 100	Flag	Night	Shout
Pos. rate		.730	.252	.083	.084	.047
AUC		.795	.837	.854	.928	.852

Fig 2.1.4. Performance of protest scene and attributes classification.

Some variables such as ‘children’ or ‘shouting’ only have a very small number of positive examples, but their model in general achieved reasonable accuracies for most variables.

Violence and Sentiment Estimation

	Violent	Angry	Fearful	Sad	Happy
Pearson’s ρ	.900	.753	.626	.340	.382
r^2	.809	.566	.392	.116	.146

Fig 2.1.5 : Performance of protest scene and attributes classification.

Their model performs very well in predicting image violence. It is less accurate for emotional sentiments. While their model successfully predicts violence and image sentiment, there are some difficult cases where the prediction does not match annotators’ ratings. The most important factor that their model does not address very well is a semantic relation between uncommon visual feature.

2.2 VGG

We found a paper on “Finding Protest in Social Media Data using CNNs and Transfer Learning” by Benjamin Zhou, Gaspar Garcia Jr, and Dylan Moore. The paper compares the performance of various classifiers in discriminating social protests from non-protests in China. With the dataset, they implemented an SVM as a baseline and CNNs. They also implemented transfer learning with VGGNet and SqueezeNet since majority of the dataset are non-protest and makes the dataset imbalanced.

2.2.1 Dataset

500,000 protest and non-protest images taken from the Chinese social media site Weibo which were gathered by Professor Jennifer Pan of Stanford’s Department of Sociology and Han Zhang, a Sociology PhD student at Princeton. The dataset consists of 231,618 protest images and 261,516 non-protest images, labeling protest images as 1 and non-protest images as 0.

Figure 2.2.1 16 sample images from the dataset they used.

2.2.2 Model

They used flattened images to fit an SVM which perfectly fitted the 20,000 training images and achieved 75% accuracy over the 2,000 sample validation set.

They used two different CNN models for the dataset. First one was 3 convolutional layers followed by two affine layers. Each convolutional layer uses relu activation and a batch normalization layer. The output from the last layer goes into a 2 by 2 max pool and it flows as described below:

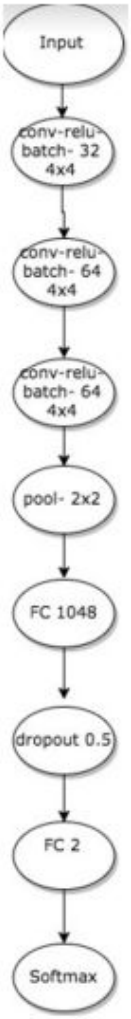

Figure 2.2.2 Architecture of 3-layer CNN

Then, they implemented 5-layer CNN. Each convolutional layer uses relu activation and a batch normalization. Architecture of 5-layer CNN is similar to 3-layer CNN except additional two layers and fully connected layer with 2048. Lastly, they implemented transfer learning by using existing neural network SqueezeNet, which was trained on ImageNet and by using VGGNet.

2.2.3 Results

Their baseline SVM classifier achieves 75% validation accuracy and 53% target accuracy. The 3-layer CNN resulted in higher validation accuracy than a deeper 5-layer CNN. Lastly, their transfer model using VGGNet achieved highest test accuracy which was 67.6%. The results are shown below:

	Train Acc.	Val Acc.	Target Acc.	Recall*	Precision*	Epochs
SVM Baseline	1	0.75	0.53	0.1	0.19	-
3-Layer CNN	0.91	0.85		0.22	0.23	15
5-Layer Deeper Neural Network	0.88	0.85	0.65	0.55	0.22	15
Transfer Learning SqueezeNet	0.89	0.86	0.62	0.72	0.24	5
Transfer Learning VGG16	0.917	0.882	0.676	0.785	0.269	5

Figure 2.2.3 Results for the various classifiers

3 Reproducing Baseline

We decided to reproduce the paper “Protest Activity Detection and Perceived Violence Estimation from Social Media Images” for our baseline. For our baseline, we implemented only the model for classifying images into classes - violence, protest and visual attributes. We did not implement the model for identifying facial emotions.

Steps Used for replicating the existing code:

1. Git clone the repository.
2. Created a virtual environment.
3. Since the Github code did not include any requirements.txt, we first analysed the code to understand the modules used.
4. Installed following modules in virtual environment:
    a. numpy
    b. pandas
    c. pillow
    d. sklearn
    e. torch
5. Their code was written for running on python2. We 7 first installed Python 2.7.15 in the virtual environment and tried running the code.
6. We ran into some errors like:

    invalid index of a 0-dim tensor. Use tensor.item() to convert a 0-dim tensor to a Python number

    We fixed the error by writing a function:

```
def __getitem__(self, index):
    if isinstance(index, slice):
        return [x**2 for x in index]
    else:
        return index**2
```

    that returns the value for 0-dim tensor.

7. We then changed the code to work with Python3 by removing Python2 dependencies like ifilter() function that required modules like itertools, and instead used Python3 built-in filter() function.

   Below is the comparison between the original model and the results we obtained:

Model	Protest Accuracy	Violence MSE	Visattr Accuracy
Original	91.7	0.00051	91.57
Frozen2	87.5	0.00444	90.00

4 Improvement

We implemented two networks for News Image classification. We used transfer learning (pre-trained weights of ImageNet dataset) and non-transfer learning to test our models.

Libraries used:

- Tensorflow
- Keras
- Numpy
- Matplotlib
- OpenCV

All models used Keras with Tensorflow backend. We did not build these models from scratch but added layers in the output of baseline model. These layers make the model and its training more robust. When transfer learning was used the weights of base model were not changed, or kept them frozen. During non-transfer learning the pretrained weights were also updated. Trained on GPU: NVIDIA GeForce GTX 1050 Ti (approx. 1.45mins/epoch) and also on BU SCC.

Why transfer learning?

Transfer learning is when we use weights of a previously trained model and then retrain/ freeze the weights to train our model. Pre-trained weights are often well trained and are good to initialize weights in our model. This helps the model to converge fast and hence, train fast in less epochs.

4.1 Added Layers

base model → Global Average Pooling → Dense(Sigmoid) → Dense(Sigmoid) → Dense(Sigmoid) → Output

4.2 Output layers

out1 = Dense(1,Sigmoid) ___violence output

out2 = Dense(1,Sigmoid) ___protest output

out3 = Dense(10,Sigmoid) ___visual attributes output

4.3 Callbacks

Callbacks were used to fine tune the model and stop training if model starts to overfit.

4.3.1 Reduce LR on Plateau

Keras uses a callback function called ReduceLROnPlateau which would reduce the learning rate if there is no change in accuracy or loss. This is often used to fine tune the hyperparameters and get slightly higher accuracy. Minimum learning rate was 0.00001.

4.3.2 Early Stopping

Early stopping is used to stop the model from training any further. We check for validation loss here, if validation loss increases or isn't changing the model stops training. This is done to prevent the model from overfitting. Patience was 3 in this case, purposely kept low as model trained in just few epochs and was very likely to overfit.

4.4 Hyperparameters

Learning rate: 0.001

Number of epochs 20

Optimizer: Adam

Loss functions:

- Mean-squared error: Violence score
- Binary Cross-entropy: Protest
- Binary Cross-entropy: Visual Attributes

4.5 ResNet-50

A standard ResNet model was used as a base model and layers were added to it. The model ran for 14 epochs and the training was stopped by Early Stopping callback.

Figure 4.5.1 Architecture of ResNet

Results:

(Transfer Learning)

Total train loss: 0.3182

Total validation loss: 1.4782

	Train loss	Validation loss	Train Accuracy	Validation Accuracy
out1	0.0118	0.0361	0.7146	0.7130
out2	0.1575	1.1626	0.9499	0.7130
out3	0.1172	0.2476	0.9606	0.9357

Test accuracy: [0.721, 0.720, 0.947]

(Non-transfer learning)

Total train loss: 0.3065

Total validation loss: 1.4432

	Train loss	Validation loss	Train Accuracy	Validation Accuracy
out1	0.0128	0.0398	0.7177	0.7075
out2	0.1513	1.1650	0.9391	0.7075
out3	0.1298	0.2618	0.9584	0.9362

Test accuracy: [0.717, 0.717, 0.9389]

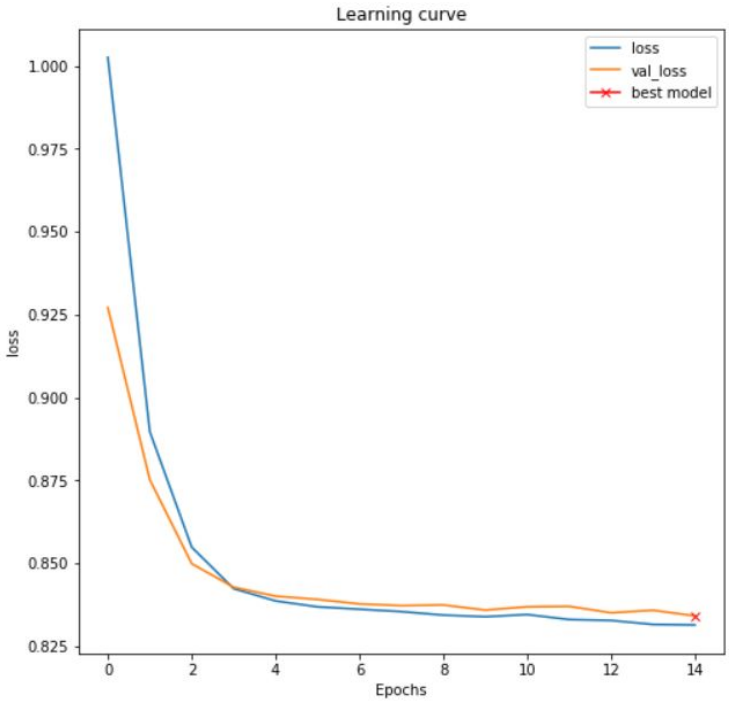

Figure 4.6.1 Learning Curve for ResNet-50

4.6 VGG-16

A standard ResNet model was used as a base model and layers were added to it. The model ran for 15 epochs.

Figure 4.6.1. Architecture of VGG-16

Results:

(Transfer learning)

Total train loss: 0.3182

Total validation loss: 1.4782

	Train loss	Validation loss	Train Accuracy	Validation Accuracy
out1	0.0332	0.0329	0.7256	0.7032
out2	0.5992	0.6203	0.7256	0.7032
out3	0.1909	0.1995	0.9366	0.9352

Test accuracy: [0.712, 0.713, 0.917]

(Non-transfer learning)

Total train loss: 0.8320

Total validation loss: 0.8346

	Train loss	Validation loss	Train Accuracy	Validation Accuracy
out1	0.0332	0.0323	0.7146	0.7239
out2	0.5992	0.6003	0.7146	0.7239
out3	0.1909	0.1935	0.9366	0.9352

Test accuracy: [0.711, 0.711, 0.929]

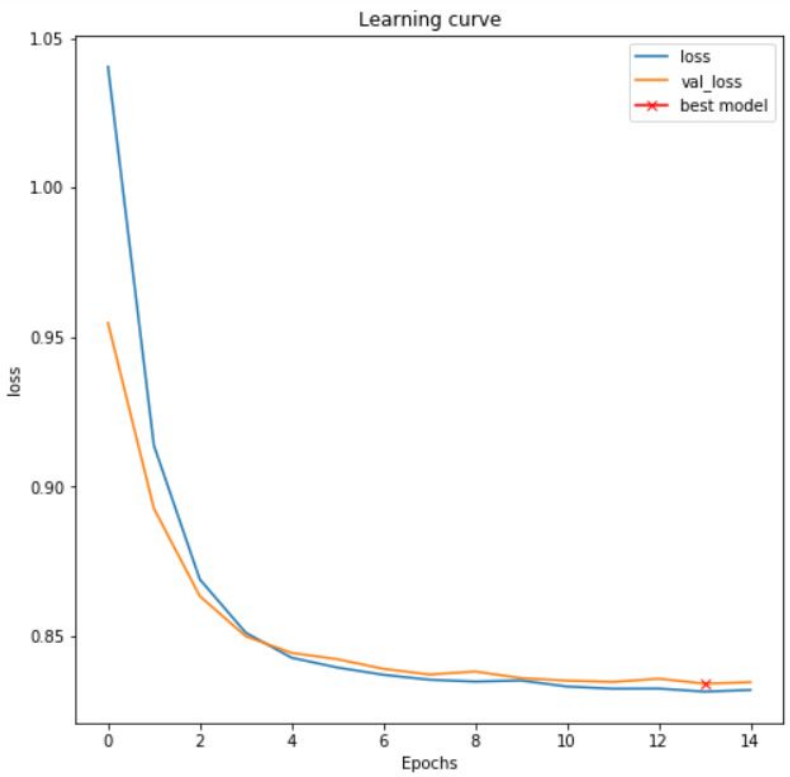

Figure 4.6.2 Learning Curve for VGG

5 Conclusion

Model implemented by us converged faster than the baseline code. Model was well trained in just 10-15 epochs while it took 100+ epochs as per the original paper. Callbacks were implemented in our model which were very useful in handling overfitting and fine tuning of parameters. Regardless of this the model didn't seem to do well on test data as it overfitted on training set.

Probable cause of Overfitting:

- Data was not good as there was a clear bias between violent and non-violent images.
- Perhaps use of bounding boxes or semantic segmentation would have given more accurate results on visual attributes. The current model is expected to learn the whole image as fire, police, shouting, etc. which it really can't achieve in multi label classification. Bounding boxes would have prepared the model to learn exactly how a label looks like.
- The model has many input features. . Effectively it could mean that the data is more sparse, so it's a lot more likely we end up drawing a conclusion that isn't warranted.

Had time permitted we would have like to do semantic segmentation for visual attributes. this would have trained the model to recognize the attributes correctly. Protest could have been detected using face recognition and then doing sentiment analysis based on facial expression. The score of sentiment analysis would give the violence score. However, we can conclude that ResNet performed better than VGG in this scenario as expected.

References

[1] (n.d.). Retrieved from https://technology.ihs.com/532501/245-million-video-surveillance-cameras-installed-globally-in-2014

[2] Zhou, B., Garcia Jr, G., & Moore, D. Finding Protests in Social Media Data using CNNs and Transfer Learning.

[3] Bruno Malveira Peixoto(2018, August 3). Violence Detection Through Deep Learning.

[4] Kwiatkowski, S., & Kwiatkowski, S. (2018, March 23). Deep Surveillance. Retrieved from https://towardsdatascience.com/deep-surveillance-6b389abeaf95

[5] Won, D., Steinert-Threlkeld, Z. C., & Joo, J. (2017, October). Protest activity detection and perceived violence estimation from social media images. In *Proceedings of the 25th ACM international conference on Multimedia* (pp. 786-794). ACM.

[6] JoshuaPiinRueyPan. (2018, August 21). JoshuaPiinRueyPan/ViolenceDetection. Retrieved from https://github.com/JoshuaPiinRueyPan/ViolenceDetection