ENEL 645 – Midterm Report

Fire and Smoke Detection using Convolutional Neural Networks

Team Members

Do Trong Anh

Bright Frimpong Anorchie

Aaron Joseph

Alan Joseph

Motivation and Significance

Detecting smoke and fire from visual scenes is a challenging task, due to the high variance of the color, texture, contrast, and sharpness. Physical solutions such as sensors, etc. have been put in place to detect fire and smoke, however they are limited in their scope and accuracy. We propose a visual based solution using deep convolutional neural networks to achieve high-accuracy detection of fire and smoke images with the eventual goal of testing it on video footage. In this midterm report we present our progress towards that goal.

Methodology

The dataset for our project was obtained from Kaggle. This dataset contained three image classes: fire, smoke and default (images without fire and smoke). All together the dataset contained 3000 images with 300 test images and 2700 training images. Further, the training data contains 900 fire images, 900 smoke images 900 default images and the test data contained 100 fire images, 100 smoke images and 100 default images.

In the first section of our notebook, we imported required deep learning packages such as NumPy, matplotlib and TensorFlow packages. Next, we imported our dataset containing train and test images. The frames for training and testing were read from the directories using ImageDataGenerator from keras. Data augmentation was performed by horizontally flipping each image by setting the horizontal flip to True in the ImageDataGenerator.
We created our model by appending fully connected layers, with "softmax" as the final classifier, to a VGG16 model (Fig 1) that is pre-trained with the weight data that comes with the dataset. This significantly reduced our training time, since the only components that needed training were the fully connected layers. In the future, we plan to add more components around this backbone to enhance the performance. We then complied this model with the standard Adam optimizer, and have the model trained in a maximum of 300 epochs and the learning rate that halves every 5 epochs.

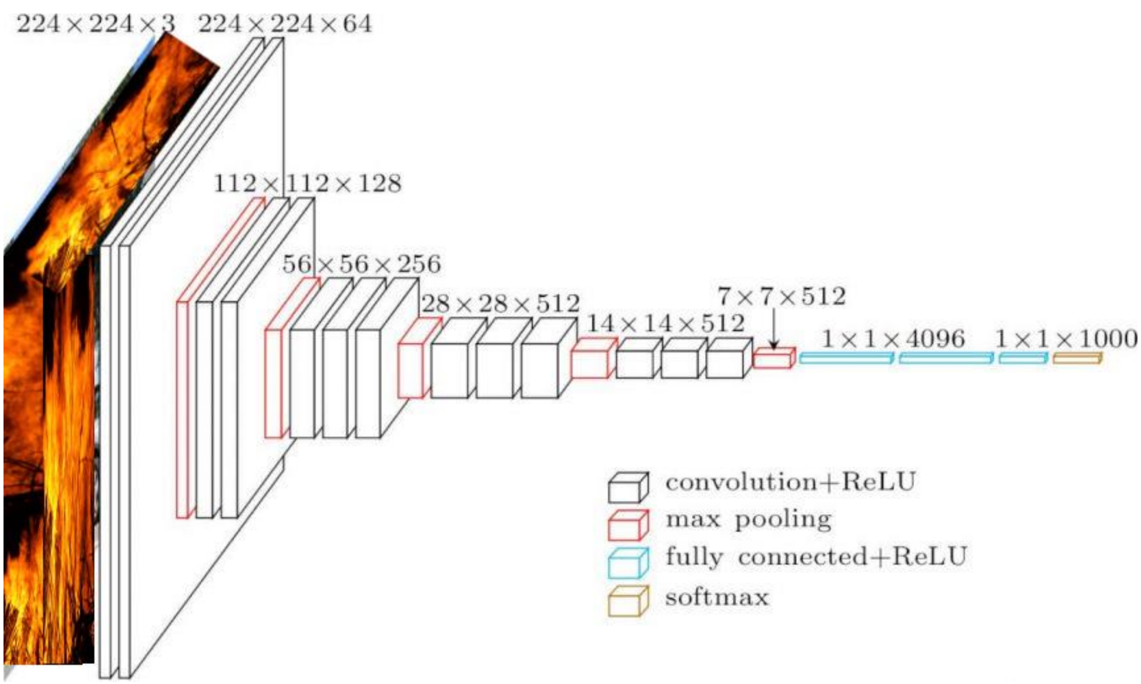

Fig 1: Representative VGG-16 model used to train our data.

Preliminary Results

We obtained the following accuracy – loss results by trying out different lengths of the fully connected components:

- Two 4096 Dense layers (ReLU), one 1000 Dense layer (ReLU) and one Softmax layer (Fig 2)
- One 4096 Dense layers, one 1000 Dense layer and one Softmax layer (Fig 4)
- One 1000 Dense layer and one Softmax layer (Fig 5)
- Only the Softmax layer (Fig 3)

We run our training with augmented data with 3 classes, training batch size of 32 and testing batch size of 1.

Fig 2: Loss results for Two 4096 Dense layers (ReLU), one 1000 Dense layer (ReLU) and one Softmax layer.

Fig 3: Loss results for only the softmax layer

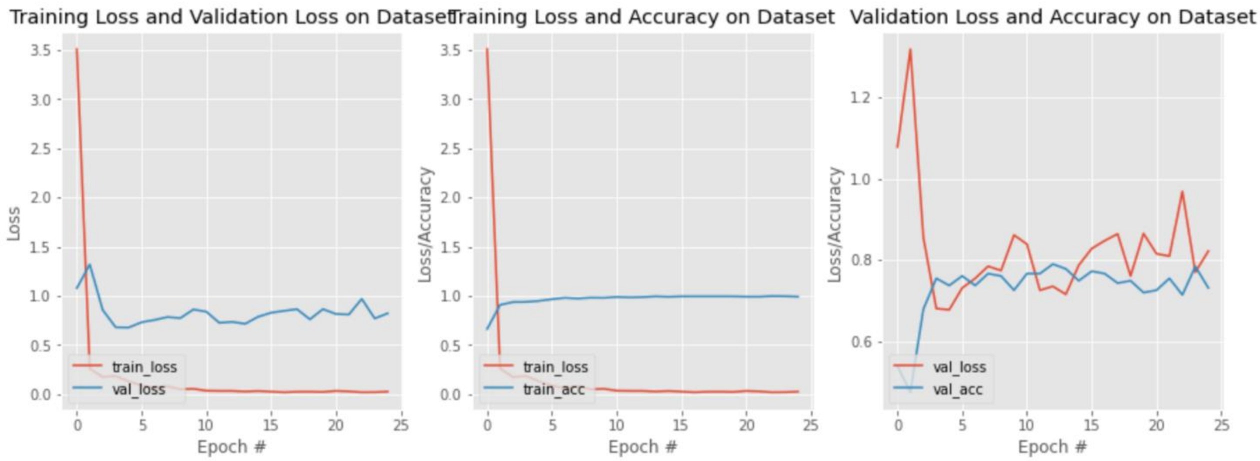

Fig 4: Loss results for One 4096 Dense layers, one 1000 Dense layer and one Softmax layer

Fig 5: One 1000 Dense layer and one Softmax layer

An important observation in the 3 graphs of each of the components above is a large gap in training-validation. The two possible reasons for this gap in terms of the dataset are:

- Dataset is not big enough
- Test set is not representative of the training set due to class imbalance.

Additionally, we can infer, that our model is not well regularized, a few possible reasons could be:

- Not enough data-points
- Too much capacity (Capacity here refers to the number of layers/nodes in the model. It seems too much capacity causes overfitting.)
- No normalization

These will be addressed as we continue to improve our model.

Future Steps

For future steps, we will try various solutions to achieve a higher performance of our VGG-16 model. We will consider the possibility of making changes to our training data. Also, since the performance of a deep learning model is constrained to the quality of its training data, we believe this can be an effective way to drive our model toward our desired results. This includes doing the following:

- Acquiring more data to further expand and train our model.
- Further rescaling and transformation of our dataset
- Using U-net model against noisy data, etc.

Additionally, we also plan to improve the performance of our deep learning model by tuning the model till we arrive at an optimal performance. Fine tuning our model may include the following:

- Experiment with exceptionally large and exceedingly small learning rates or try out a learning rate that decreases over epochs.
- Experiment with different activation functions

The video that will be used finally for frame-by-frame classification testing will be CCTV footage (224x224) of several fire incidents in urban areas. The fire(s) will be in full view, generating smoke, with which our model will then test every frame to determine which frame contains fire, smoke, both fire and smoke or neither.

References

1. A. Jadon, Mohd. Omama, Akshay Varshney, “FireNet: A specialized lightweight fire & smoke detection Model for Real-time IoT Applications,” 20 May 2019.
2. Feiniu Yuan, Lin Zhang, Xue Xia, “Deep Smoke Segmentation,” 28 August 2018.
3. Abdulaziz Namozov, “An Efficient Deep Learning Algorithm for Fire and Smoke Detection with limited data,” Vol 8, Number 4, 2018.
4. Çelik, Turgay, Hüseyin Özkaramanlı, and Hasan Demirel. "Fire and smoke detection without sensors: Image processing-based approach." In 2007 15th European Signal Processing Conference, pp. 1794-1798. IEEE, 2007.
5. C. Tao, J. Zhang, and P. Wang, "Smoke Detection Based on Deep Convolutional Neural Networks," 2016 International Conference on Industrial Informatics – Computing Technology, Intelligent Technology, Industrial Information Integration (ICIICII), Wuhan,China, 2016
6. Khan Muhammad, Jamil Ahmad, Sung Wook Baik, “Early fire detection using convolutional neural networks during surveillance for effective disaster management”, Neurocomputing, Volume 288, 2018, ISSN 0925-2312
7. https://www.pyimagesearch.com/2019/11/18/fire-and-smoke-detection-with-keras-and-deep-learning/
8. Frizzi, Sebastien & Kaabi, Rabeb & Bouchouicha, Moez & Ginoux, Jean-Marc & Moreau, Eric & Fnaiech, Farhat. (2016). Convolutional neural network for video fire and smoke detection. 877-882. 10.1109/IECON.2016.7793196.
9. NAMOZOV, A. & CHO, Y.. (2018). An Efficient Deep Learning Algorithm for Fire and Smoke Detection with Limited Data. Advances in Electrical and Computer Engineering. 18. 121-128. 10.4316/AECE.2018.04015.
10. Dua, Mohit & Kumar, Mandeep & Charan, Gopal & Ravi, Parre. (2020). An Improved Approach for Fire Detection using Deep Learning Models. 171-175. 10.1109/I4Tech48345.2020.9102697.
11. D. Kim and Y. Wang, "Smoke Detection in Video," 2009 WRI World Congress on Computer Science and Information Engineering, Los Angeles, CA, USA, 2009, pp. 759-763, doi: 10.1109/CSIE.2009.494.
12. E. liu, "Research on Video Smoke Recognition Based on Dynamic Image Segmentation Detection Technology," 2019 12th International Conference on Intelligent Computation Technology and Automation (ICICTA), Xiangtan, China, 2019, pp. 240-243, doi: 10.1109/ICICTA49267.2019.00058.
13. P. Ma, F. Yu, C. Zhou and M. Jiang, "An Integrated Smoke Detection Method based on Convolutional Neural Network and Image Processing," 2020 IEEE 8th International Conference on Computer Science and Network Technology (ICCSNT), Dalian, China, 2020, pp. 32-36, doi: 10.1109/ICCSNT50940.2020.9304985.
14. Y. Zhang and Y. Hu, "Video Smoke Detection Based on Convolution Neural Network," 2017 International Conference on Computer Technology, Electronics and Communication (ICCTEC), Dalian, China, 2017, pp. 1334-1338, doi: 10.1109/ICCTEC.2017.00293.

15. Kim, B.; Lee, J. A Video-Based Fire Detection Using Deep Learning Models. *Appl. Sci.* 2019, *9*, 2862. https://doi.org/10.3390/app9142862

Team Member's Effort and Scores

- Do Trong Anh: Model creation and Design, Dataset preprocessing and report writing.
- Bright Anorchie: Model creation and Design, Dataset preprocessing and report writing.
- Aaron Joseph: Model creation and Design, Dataset preprocessing and report writing.
- Alan Joseph: Model creation and Design, Dataset preprocessing and report writing.

Team Members & Score

Name	**UCID**	**Activity**	**Score**
Bright Anorchie	30108981	Model creation and Design, Dataset preprocessing and report writing.	3
Do Trong Anh	30111540	Model creation and Design, Dataset preprocessing and report writing.	3
Aaron Joseph	30109249	Model creation and Design, Dataset preprocessing and report writing.	3
Alan Joseph	30121097	Model creation and Design, Dataset preprocessing and report writing.	3