Human Suspicious Activity Detection using Deep Learning

Rachana Gugale[1], Abhiruchi Shendkar[2], Arisha Chamadia[3], Swati Patra[4], Deepali Ahir[5]

[1,2,3,4]Student, Department of Computer Engineering, M. E. S. College of Engineering, Pune, India
[5]Assistant Professor, Department of Computer Engineering, M. E. S. College of Engineering

---***---

Abstract - *Detecting suspicious activities in public places has become an important task due to the increasing number of shootings, knife attacks, terrorist attacks, etc. happening in public places all around the world. This paper focuses on a deep learning approach to detect suspicious activities using Convolutional Neural Networks from images and videos. We analyze different CNN architectures and compare their accuracy. We give the architecture of our system which can process video footage in real time from cameras and predict if the activity is suspicious or not. We also propose future developments which can be made in this area of suspicious activity detection.*

Key Words: Suspicious Activity Detection, Convolutional Neural Networks, FastAI, Deep Learning

1. INTRODUCTION

Suspicious human activity recognition from surveillance video is an active research area of image processing and computer vision. Through the visual surveillance, human activities can be monitored in sensitive and public areas such as bus stations, railway stations, airports, banks, shopping malls, school and colleges, parking lots, roads, etc. to prevent terrorism, theft, accidents and illegal parking, vandalism, fighting, chain snatching, crime and other suspicious activities. It is very difficult to watch public places continuously, therefore an intelligent video surveillance is required that can monitor the human activities and categorize them as usual and unusual activities; and can generate an alert.

1.1 Previous Approaches

For detecting suspicious human activity, it is important for the model to learn suspicious human poses. Human pose estimation is one of the key problems in computer vision that has been studied for more than 15 years. It is related to identifying human body parts and possibly tracking their movements. It is used in AR/VR, gesture recognition, gaming consoles, etc. Initially, low cost depth sensors (motion sensors) were used to find human movement in gaming consoles. However, these sensors are limited to indoor use, and their low resolution and noisy depth information make it difficult to estimate the human activity going on from depth images. Hence, they are not a suitable option for suspicious activity detection. Models like OpenPose[1], PoseNet[2] give out the keypoint coordinates of the people in the image/video in real time. But just obtaining the keypoints of the people without any background or surrounding objects information is not enough to decide if an activity is suspicious. So, we use a CNN approach in our system instead of using a keypoints based approach.

2. METHODOLOGY

The first step was to decide which suspicious activities to focus on. We selected 5 suspicious activities to classify: Shooting, punching, kicking, knife attack and sword fight. These 5 activities formed 5 classes for our classifier model. The non-suspicious activities were put in a 6th class.

The next step was to collect data for each of the classes. Images were scraped from Google Images by using a JavaScript code snippet. Once we collected enough images, we manually filtered the irrelevant images. This process was repeated for each of the 6 classes. The total number of images in our dataset is 17,716.

Once we had our data, we started the process of model selection. After researching about neural network model architectures and which ones to use for real-time tasks, we decided to use ResNet. We decided to experiment with ResNet-18, 34 and 50. The numbers here stand for the number of neuron layers in the model architecture.
For training and evaluating the model, we used the deep learning framework FastAI[3],[4] which is based on PyTorch[5]. FastAI is organized around two main design goals: to be approachable and rapidly productive, while also being deeply hackable and configurable. It has the clarity and development speed of Keras[6] and the customizability of PyTorch. This goal of getting the best of both worlds has motivated the design of a layered architecture for FastAI. A high-level API powers ready-to-use functions to train models in various applications, offering customizable models with sensible defaults. The FastAI APIs choose intelligent default values and behaviors based on all available information. For instance, FastAI provides a single Learner class which brings together architecture, optimizer, and data, and automatically chooses an appropriate loss function where possible. The use of intelligent defaults – based on FastAI creators' experience or best practices – extends to incorporating state-of-the-art research wherever possible. For instance, transfer learning is critically important for training models quickly, accurately, and cheaply, but the details matter a great deal. FastAI

automatically provides transfer learning, optimized batch-normalization, training, layer freezing, and discriminative learning rates. In general, the library's use of integrated defaults means it requires fewer lines of code from the user to re-specify information or merely to connect components. As a result, every line of user code tends to be more likely to be meaningful, and easier to read.

FastAI APIs were used to divide the dataset into training and validation set. 20% of the data was used for validation while the rest of 80% was used for training.

3. SYSTEM ARCHITECTURE

The system is divided into 2 phases: training and deployment. During the training phase, the ResNet model is trained with our custom dataset. We divide the dataset into training and validation sets. Validation set contained 20% of the images which were randomly chosen from the dataset.

After the training phase, the model is deployed on computer systems used by the security teams in public places. Our system is a desktop application which can take as input live feed from a camera or an already stored video from the computer. This video is then preprocessed (which involves breaking the video into frames) and then fed into the ResNet-50 model. The model outputs if the video contains any suspicious activity or not. If a suspicious activity is detected in the video, the model immediately generates an alert on the system and also sends an email alert on the registered email address along with pictures of the video where suspicious activity is ongoing.

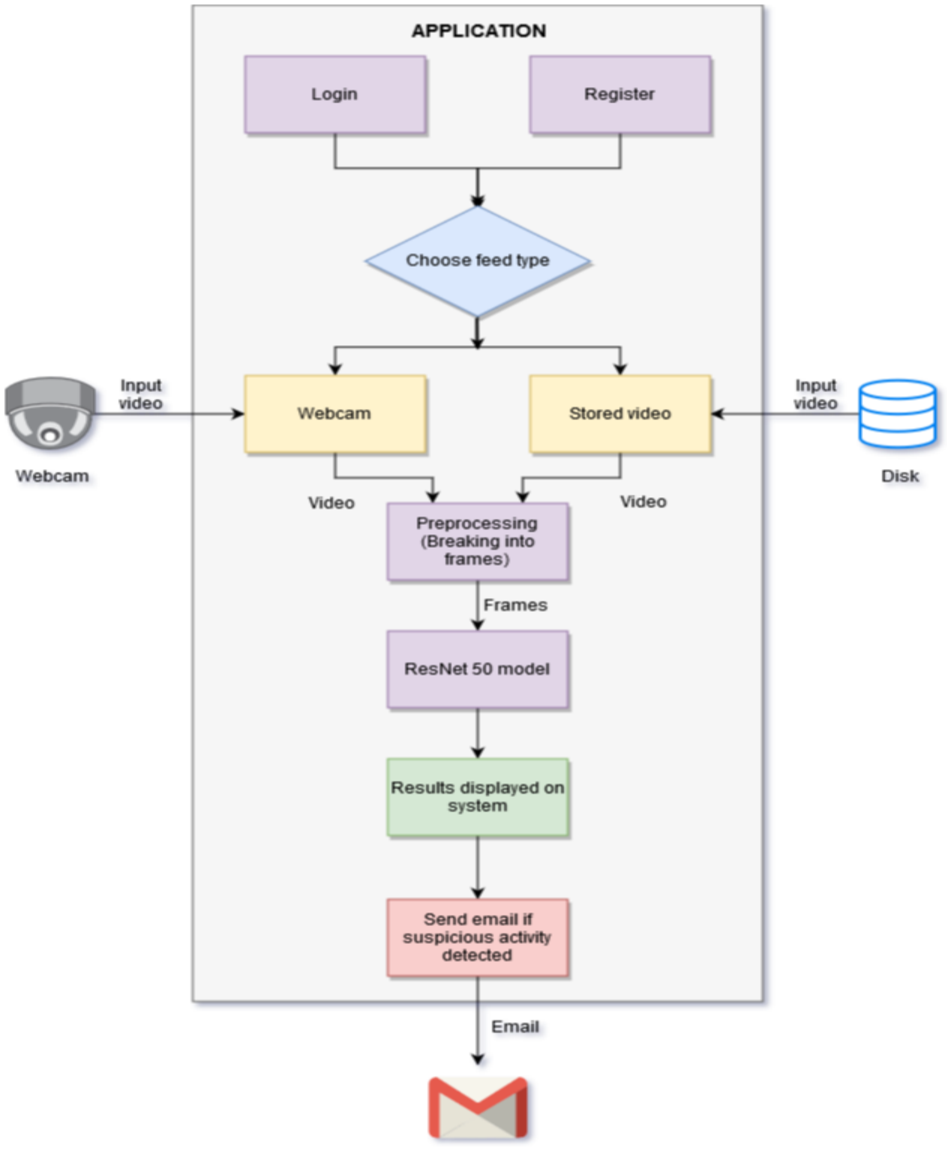


Figure 1: System Architecture

4. RESULTS

Suspicious activity detection has become an important area of study due to the increasing number of crimes happening. We studied the previous approaches present and offered an alternative approach to detect suspicious activities happening in public places. Our approach used CNN for finding if the activity was suspicious. The ResNet architecture was used to build the CNN model. We tried ResNet-18, ResNet-34 and ResNet-50 approaches. We also tried to train our model with default learning rate and learning rate in the range of 3×10^(-5) to 3×10^(-4). The following results were obtained for each architecture:

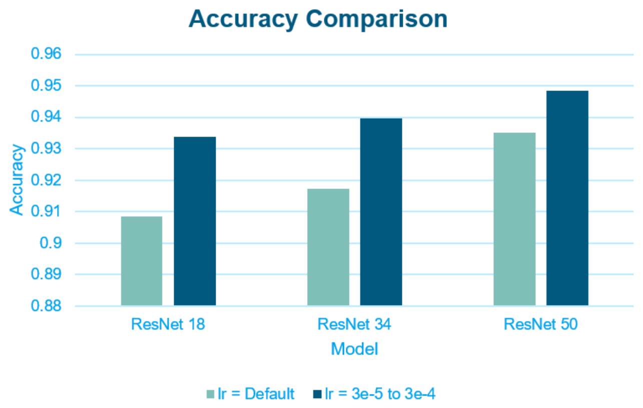
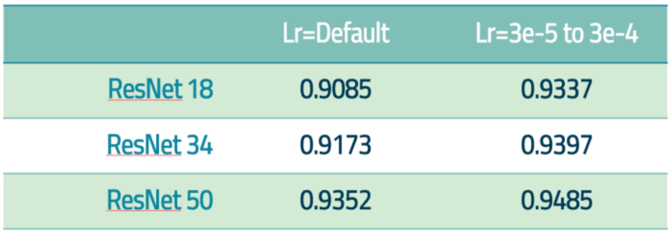
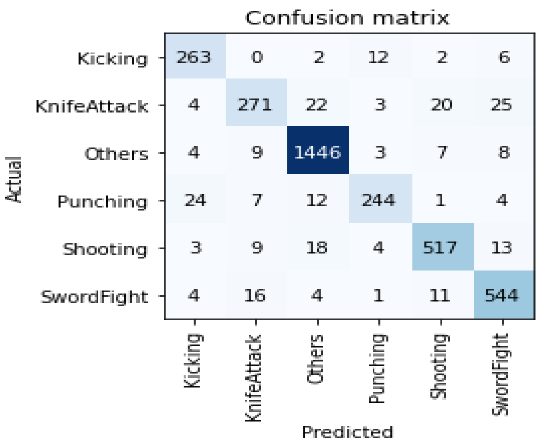


Figure 2: Accuracy Comparison Graph



	Lr=Default	Lr=3e-5 to 3e-4
ResNet 18	0.9085	0.9337
ResNet 34	0.9173	0.9397
ResNet 50	0.9352	0.9485

Figure 3: Obtained Accuracies



	Kicking	KnifeAttack	Others	Punching	Shooting	SwordFight
Kicking	263	0	2	12	2	6
KnifeAttack	4	271	22	3	20	25
Others	4	9	1446	3	7	8
Punching	24	7	12	244	1	4
Shooting	3	9	18	4	517	13
SwordFight	4	16	4	1	11	544

Figure 4: Obtained Confusion Matrix

5. CONCLUSION

According to the results obtained by the above architectures, we conclude that ResNet-50 works the best for this task. A learning rate in the range of 3×10^(-5) to 3×10^(-4) also works the best.

ACKNOWLEDGEMENTS

It gives us great pleasure and satisfaction to have worked on “Human Suspicious Activity Detection using Deep Learning”. We are thankful to and fortunate enough to get constant encouragement, support, and guidance from our guide Prof. Deepali Ahir. She encouraged us to keep moving forward under her guidance and vigilant support. We would also like to extend our sincere thanks to all friends and family for their motivation in times that we hit a wall. We would like to thank all those, who have directly or indirectly helped us for the completion of the work during this project.

FUTURE AREAS OF RESEARCH

Our model currently targets only 5 suspicious activities. It can further be improved by targeting a greater number of suspicious activities. More images can be added to the current dataset, especially images extracted from the CCTV footage of the suspicious activity. Such footage is currently difficult to obtain as students but if this project is supported by the civic administration, they can surely provide the footage of criminal activities which have happened over this past. This will vastly help in improving the model.

REFERENCES

[1] Cao, Zhe, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2018. "OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields." arXiv.

[2] Cipolla, Alex Kendall, Matthew Grimes, and Roberto. 2015. "PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization." arXiv.

[3] Howard, Jeremy, and S Gugger. 2020. Deep Learning for Coders with fastai and PyTorch: AI Applications Without a PhD. O'Reilly Media, Inc.

[4] Howard, Jeremy, and Sylvain Gugger. 2020. "fastai: A Layered API for Deep Learning." arXiv.

[5] Paszke, Adam, Sam Gross, and Others. 2019. "PyTorch: An Imperative Style, High-Performance Deep Learning Library." In Advances in Neural Information Processing Systems 32, 8024--8035. Curran Associates, Inc.

[6] Chollet, François. 2015. "Keras." GitHub repository (GitHub). https://github.com/fchollet/keras.