Threat Detection in Surveillance Cameras using Deep Learning and Transfer Learning

MINOR PROJECT REPORT

Submitted in partial fulfilment of the requirements for the award of

the degree of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

by

Jai Sharma	**Muskaan Patel**	**Utsav Garg**
Enrollment No: 35211503117	**Enrollment No: 40711503117**	**Enrollment No: 35611503117**

Guided by

Mr. Anurag Agarwal

Assistant Professor

DEPARTMENT OF INFORMATION TECHNOLOGY
BHARATI VIDYAPEETH'S COLLEGE OF ENGINEERING
(AFFILIATED TO GURU GOBIND SINGH INDRAPRASTHA UNIVERSITY, DELHI) DELHI – 110063

DECEMBER 2020

Threat Detection in Surveillance Cameras using Deep Learning and Transfer Learning

MINOR PROJECT REPORT

Submitted in partial fulfilment of the requirements for the award of

the degree of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

by

Jai Sharma	**Muskaan Patel**	**Utsav Garg**
Enrollment No: **35211503117**	**Enrollment No:** **40711503117**	**Enrollment No:** **35611503117**

Guided by

Mr. Anurag Agarwal

Assistant Professor

DEPARTMENT OF INFORMATION TECHNOLOGY
BHARATI VIDYAPEETH'S COLLEGE OF ENGINEERING
(AFFILIATED TO GURU GOBIND SINGH INDRAPRASTHA UNIVERSITY, DELHI) DELHI – 110063

DECEMBER 2020

CANDIDATE’S DECLARATION

It is hereby certified that the work which is being presented in the B. Tech Major Project Report entitled “**Threat Detection in Surveillance Cameras using Deep Learning and Transfer Learning"** in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** and submitted in the **Department of Information & Technology Engineering** of **BHARATI VIDYAPEETH’S COLLEGE OF ENGINEERING, New Delhi (Affiliated to Guru Gobind Singh Indraprastha University, Delhi)** is an authentic record of our own work carried out during a period from **May 2020 to December 2020** under the guidance of **Mr. Anurag Agarwal, Assistant Professor.**

The matter presented in the B. Tech Minor Project Report has not been submitted by me for the award of any other degree of this or any other Institute.

(Jai Sharma)	**(Muskaan Patel)**	**(Utsav Garg)**
(En. No:35211503117)	**(En. No:40711503117)**	**(En. No: 35611503117)**

This is to certify that the above statement made by the candidate is correct to the best of my knowledge. He/She/They are permitted to appear in the External Minor Project Examination.

(Mr. Anurag Agarwal)
Assistant Professor

(Dr. Prakhar Priyadarshani)
Head, IT Dept.

The B. Tech Minor Project Viva-Voce Examination of **Muskaan Patel (Enrollment No: 40711503117),** has been held on **……………………………….**

Project Coordinator **Project Coordinator** **(Signature of External Examiner)**

ABSTRACT

Closed circuit television systems (CCTV) are becoming more and more popular and are being deployed in many offices, housing estates and in most public spaces. Thus, the job of CCTV operators is becoming very challenging as the footage contains a lot of information and gradually becomes cumbersome. Rapid advancement in the field of computer vision could be observed as an important trend in video surveillance and lead to substantial efficiency gains. Through this system of a neural network based intelligent intruders' detection and tracking system using Closed-Circuit Television (CCTV) images, we can examine the techniques and algorithms used to identify a potential intruder and methods to eliminate other non-threatening objects. We use image processing and convolutional neural networks (CNN) and transfer learning to detect any anomalous activity in CCTV footage, alleviating the waste of labor and time.

ACKNOWLEDGEMENT

We express our deep gratitude to **Mr. Anurag Agarwal**, Assistant Professor, Department of Information & Technology Engineering for his valuable guidance and suggestion throughout my project work. We are thankful to **Mrs. Neha Gupta and Mrs. Sarita Yadav,** Class Advisors, for their valuable guidance.

We would like to extend my sincere thanks to **Head of the Department, Dr. Prakhar Priyadarshani** for his time-to-time suggestions to complete my project work. I am also thankful to **Dr. Dharmender Saini, Principal** for providing me the facilities to carry out my project work.

Sign	**Sign**	**Sign**
(Jai Sharma)	**(Muskaan Patel)**	**(Utsav Garg)**
(En. No:35211503117)	**(En. No:40711503117)**	**(En. No:35611503117)**

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

CNN	Convolution Neural Network
SVM	Support Vector Machine
3D	Three Dimensional
CCTV	Closed Circuit Television
DL	Deep Learning
CV	Computer Vision
R-CNN	Regional Convolutional Neural Network
IDC	International Data Conference
OpenCV	Open Computer Vision Library
RGB	Red, Green and Blue
FC	Fully Connected
OID	Open Image Dataset
API	Application Programming Interface
NLP	Natural Language Processing

CHAPTER 1: INTRODUCTION

1.1 INTRODUCTION

Video surveillance systems have been introduced in various fields of our daily life to enhance security and protect individuals and sensitive infrastructure. Object detection and tracking has wide applications such as people tracking, safety monitoring, security and bio metrics, traffic and road management, web applications, object recognition for mobile devices and others.

Video surveillance pays a great role in the field of robbery detection. In today's generation of rapid technological advancement, the demand of automatic video surveillance systems has been increasing exponentially.

Due to growing poverty and population rate 1.12 % per year (the average increase is eighty-three million per year), crime rate is also increasing. Due to this growing rate of street crimes in our city, the surveillance cameras have been installed inside and outside almost all the public places. Thus, security industry is also growing all around the world, exponentially.

CCTV are winding up increasingly famous and are deployed in and around all the public places. In the past few years, the multiple CCTV cameras were installed for surveillance purposes, but the monitoring purpose of all these cameras became a challenging task for everyone.

The traditional method CCTV security systems which rely solely upon a human operator to perform such tasks as object classification and analysis, have some inherent limitations. The first is that the successful operation of such a security system depends on the reliability of the human operator. It has been shown that the average attention span of a human operator in an environment such as that used for typical CCTV monitoring applications can be as short as 20 minutes. After this time, movement taking place within the frame of view of a CCTV camera could be unnoticed by the human operator. Also, there are some situations where visual monitoring of the camera scenes may be required for extended periods, in which case a human operator may prove to be unreliable as a result of short attention spans and fatigue.

A solution to combat these issues is to apply image-processing and CV algorithms to the CCTV footage, which will eliminate the need of human operatives and would automatically alert the security officials if a dangerous situation occurs. When a person holds a weapon out in the open, it indicates a possibly dangerous situation. However, some countries allow people to carry firearms freely and openly, but even in such cases, it is wise to grasp the attention of CCTV operatives in order to evaluate such conditions or scenarios. In the previous few years, Deep Learning (DL) and Convolutional Neural Networks (CNN's) have accomplished best results to all the classical machine learning methods in image detection, classification, etc.

Using CNN’s, we can focus on building a good weapon detector in real time. The proposed system focuses on implementing an Intelligent Surveillance Camera which monitors the activity recorded by the CCTV, and detects any type of suspicious weapon, behavior or person. If any such activity is detected, the software will automatically send an alert message to the security department.

The proposed methodology uses R-CNNs (Regional-Convolutional Neural Network). To bypass the problem of selecting a huge number of regions, Ross Girshick et al. proposed a method where we use selective search to extract just 2000 regions from the image and he called them region proposals. Therefore, now, instead of trying to classify a huge number of regions, you can just work with 2000 regions. These 2000 region proposals are generated using the selective search algorithm.Deep convolutional neural network models may take days or even weeks to train on very large datasets.

A way to short-cut this process is to re-use the model weights from pre-trained models that were developed for standard computer vision benchmark datasets, such as the ImageNet image recognition tasks. Top performing models can be downloaded and used directly, or integrated into a new model for your own computer vision problems.

Transfer learning involves using models trained on one problem as a starting point on a related problem. Transfer learning is flexible, allowing the use of pre-trained models directly, as feature extraction preprocessing, and integrated into entirely new models. Keras provides convenient access to many top performing models on the ImageNet image recognition tasks such as VGG, Inception, and ResNet.

Transfer learning generally refers to a process where a model trained on one problem is used in some way on a second related problem. In deep learning, transfer learning is a technique whereby a neural network model is first trained on a problem similar to the problem that is being solved. One or more layers from the trained model are then used in a new model trained on the problem of interest.

This is typically understood in a supervised learning context, where the input is the same but the target may be of a different nature. For example, we may learn about one set of visual categories, such as cats and dogs, in the first setting, then learn about a different set of visual categories, such as ants and wasps, in the second setting.

Its main objective is to detect three types of suspicious activities and alert the person monitoring the surveillance footage by sending either a message/short audio clip:
1. Any harmful/dangerous weapons like gun, bullet, rifle, etc.
2. Any threatening motion/activity like hands up, people lying down on the
floor, etc.
3. Any objectionable/off-putting accessory like a complete face mask, etc.

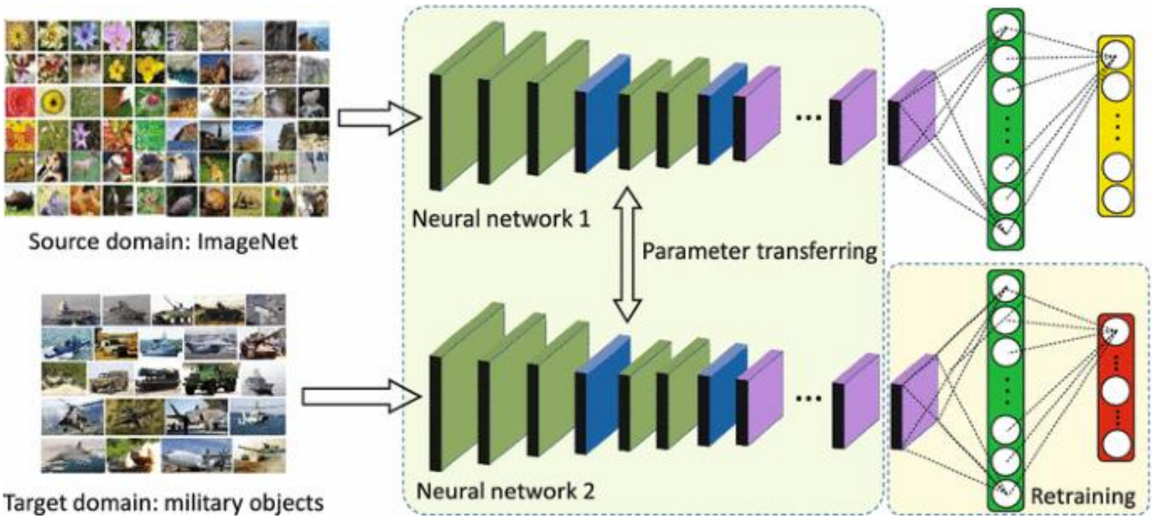


Figure 1: Application of Transfer Learning in Object Detection

Transfer learning has the benefit of decreasing the training time for a neural network model and can result in lower generalization error. The weights in re-used layers may be used as the starting point for the training process and adapted in response to the new problem. This usage treats transfer learning as a type of weight initialization scheme. This may be useful when the first related problem has a lot more labeled data than the problem of interest and the similarity in the structure of the problem may be useful in both contexts.

A challenge, often referred to simply as ImageNet, given the source of the image used in the competition, has resulted in a number of innovations in the architecture and training of convolutional neural networks. In addition, many of the models used in the competitions have been released under a permissive license.

These models can be used as the basis for transfer learning in computer vision applications. This is desirable for a number of reasons, not least:

1. Useful Learned Features: The models have learned how to detect generic features from photographs, given that they were trained on more than 1,000,000 images for 1,000 categories.

2. State-of-the-Art Performance: The models achieved state of the art performance and remain effective on the specific image recognition task for which they were developed.

3. Easily Accessible: The model weights are provided as free downloadable files and many libraries provide convenient APIs to download and use the models directly. The model weights can be downloaded and used in the same model architecture using a range of different deep learning libraries, including Keras.

The use of a pre-trained model is limited only by your creativity. For example, a model may be downloaded and used as-is, such as embedded into an application and used to classify new photographs.

1.2 MOTIVATION

All over the world, security industry is growing at a rapid rate. In particular, more and more video surveillance systems have been installed for the monitoring of public places and private premises. While most of these systems require human operators to monitor the CCTV images at a centralized location, studies have shown that the operators suffer from a rapid loss of concentration once fatigue sets in. In addition, they have limited capability to monitor more than a few cameras at any given time. It is therefore desirable to have an automated system which does not suffer from these limitations.

Due to the growing rate of street crimes in our city; the surveillance cameras should be installed inside and outside the public places. The surveillance cameras help people to stay alert by detecting an unethical activity within a second, and the surveillance cameras can be useful for further investigation purposes such as tracking an identity of a criminal or a victim. Although, the surveillance cameras are being used in public offices to monitor the ethical and unethical behavior of their employees or their visitors for instance.

However, the growing demand for CCTV cameras used for surveillance purposes is stressful for people who operate multiple screens of these CCTV cameras. In the past few years, the multiple CCTV cameras were installed for surveillance purposes, but the monitoring purpose of all these cameras became a challenging task for everyone.

For instance, the number of CCTV screens were increased but the concentration of human on each screen was decreased. Therefore, since after a growing demand for CCTV cameras for surveillance purposes the authorities hired people for working on automated surveillance algorithms.

One way to reducing this kind of violence is prevention via early detection so that the security agents or policemen can act. In particular, one innovative solution to this problem is to equip surveillance or control cameras with an accurate automatic handgun detection alert system. Related studies address the detection of guns but only on X-ray or millimetric wave images and only using traditional machine learning methods.

In the last five years, deep learning in general and Convolutional Neural Networks (CNNs) in particular have achieved superior results to all the classical machine learning methods in image classification, detection and segmentation in several applications. Instead of manually selecting features, deep learning CNNs automatically discover increasingly higher level features from data. We aim at developing a good gun detector in videos using CNNs.

To satisfy the actual demands, a lot of places have been installed with the smart cameras. Most of those products can provide pre-alarming functions for these special occasions, like banks. However, there are some limitations in using these smart cameras for detecting abnormal events:

(1) The existing intelligent surveillance systems can only detect and alarm single abnormal event yet without bridging the spatial and temporal association among multiple unusual events. However, it is quite not convincible to judge suspicious behavior by a single monitoring. As the case of wandering in the front of a bank, the occasional wander outside the bank may be a usual behavior for awaiting others. It only makes sense to treat the wandering as suspicion when it happens repeatedly or takes a long time.

(2) The huge amount of video acquired by the city scale monitoring network results in the rapid increasing of storage costs. IDC (International Data Corporation) calculated that surveillance video data accounts for 65 percent of whole data, which was far more than any other data like transaction data, medical data, entertainment and social media data. Since the massive surveillance video needs to be stored for several months or years, it leads to a large storage cost.

(3) The amount of false alarming resulted from the data explosion is beyond the limitations of manual processing. Traditional methods for obtaining evidences highly depend on the surveillance video within or near the accident site. However, when the incident passes through a wide range of space and time, it is hard to find any valuable evidences on the criminals from massive surveillance video, which hampers the efficiency of resolving cases.

The recent emerging smart monitoring cameras are able to automatically identify abnormal behaviors through the built-in intelligent algorithms, greatly boosting the performance of the surveillance system. However, the above mentioned three major challenges have not been fundamentally resolved. The essential reason is that the existing system only individually accepts alarm information from each front-end camera and makes a limited range of alarming, without performing collaborative analysis among geospatially interrelated camera network. Besides, the detection results on unusual behaviors are not fully exploited in terms of deep utilization, paying little attention to storage and retrieval on massive video but for event alarming.

1.3 OBJECTIVE

Its main objective is to detect any harmful/dangerous weapons like gun, bullet, rifle, etc and alert the person monitoring the surveillance footage by sending either a message/short audio clip.

This paper proposes the application of Transfer Learning in order to make our deployed API model more specific to our project. Currently, the deployed API model is using the Inception RESNET v2 architecture on the ImageNet dataset.

We have further applied Transfer Learning to our pre-trained model in order to make the model more specific to our required dataset. We extracted our dataset from the Google's Open Image Dataset v6 (OIDv6). We extracted images from 11 different subclasses and trained our personalized model on that dataset to increase the efficiency and accuracy of our own model.

1.4 SUMMARY OF THE REPORT

The report on the topic “Threat Detection in Surveillance Cameras using Deep Learning and Transfer Learning” has in total 5 chapters. The first chapter has 4 subchapters. This report also contains Introduction, Acknowledgement, Candidate’s Declaration along with the List of Tables, Figures and Abbreviation. The report ends with the list of references.

The first chapter, i.e. Introduction, includes 4 subchapters with the following names: Introduction, Motivation, Objective and Summary of the Report. The Introduction part contains an overview of the topic, its current scenario in the country. The Motivation subchapter deals with the fact as to why the author chose this topic for their project, what motivated them to do so. The third subchapter, Objective, informs the reader about what does the project exactly do, what is its purpose in the form of bullet points. Lastly, Summary of the Report provides the reader with a small summarized version of the full project’s report.

The second chapter, i.e. Methodology gives a detailed description about the working of the project. It tells about the various methodologies used and a small description about the working of the methodology as well. It also tells us about the dataset used, etc. It also gives a deep insight about Transfer Learning and its various applications.

The third chapter, i.e. Results and Discussion, gives the user about the result of the project. It includes the accuracy and figures of our trained model.

Lastly, the fourth chapter, i.e. Conclusion, concludes the whole report and gives a short summary of the whole report of the project. It also discussed the future work for the project implementation.

CHAPTER 2: METHODOLOGY

Our methodology for the construction of an Intelligent Surveillance Camera that detects anomalous activity/person/behavior is as follows:

Step 1: Video to Frame Conversion:

Firstly, the input video is converted into frames using OpenCV in Python. OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products.

The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc. OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 18 million. The library is used extensively in companies, research groups and by governmental bodies.

Step 2: Downloading our Pre-Trained Model:

The pre-trained model we are using is that called Inception RESNET v2. Inception ResNet V2 is a neural network architecture for image classification, originally published by Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, Alex Alemi: "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning", 2016.

Its weights were originally obtained by training on the ILSVRC-2012-CLS dataset for image classification ("ImageNet"). Inception_Resnet_v2 is formulated based on a combination of the Inception structure and the Residual connection. In the Inception-Resnet block, multiple sized convolutional filters are combined by residual connections.

The usage of residual connections not only avoids the degradation problem caused by deep structures but also reduces the training time. Inception-ResNet-v2 is a convolutional neural network that is trained on more than a million images from the ImageNet database. The network is 164 layers deep and can classify images into 1000 object categories, such as the keyboard, mouse, pencil, and many animals. As a result, the network has learned rich feature representations for a wide range of images. The network has an image input size of 299-by-299, and the output is a list of estimated class probabilities.

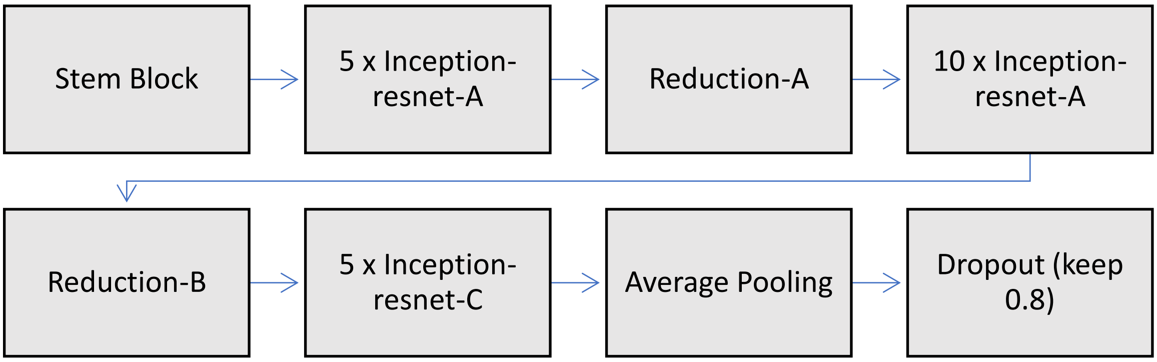


Figure 2: Inception RESNET v2 Architecture [5]

A pre-trained model has been previously trained on a dataset and contains the weights and biases that represent the features of whichever dataset it was trained on. Learned features are often transferable to different data. For example, a model trained on a large dataset of bird images will contain learned features like edges or horizontal lines that you would be transferable to your dataset.

ResNet and Inception have been central to the largest advances in image recognition performance in recent years, with very good performance at a relatively low computational cos t. Inception-ResNet combines the Inception architecture, with residual connections.

Each Inception block is followed by a filter expansion layer
(1 × 1 convolution without activation) which is used for scaling up the dimensionality of the filter bank before the addition to match the depth of the input.
2. In the case of Inception-ResNet, batch-normalization is used only on top of the traditional layers, but not on top of the summations.

If the number of filters exceeded 1000, the residual variants started to exhibit instabilities and the network has just “died” early in the training, meaning that the last layer before the average pooling started to produce only zeros after a few tens of thousands of iterations. This could not be prevented, neither by lowering the learning rate nor by adding an extra batch normalization to this layer.

According to them, scaling down the residuals before adding them to the previous layer activation seemed to stabilize the training. To scale the residuals, scaling factors between 0.1 and 0.3 were picked.

Inception-ResNet-v2 is a convolutional neural network that is trained on more than a million images from the ImageNet database. The network is 164 layers deep and can classify images into 1000 object categories, such as the keyboard, mouse, pencil, and many animals. As a result, the network has learned rich feature representations for a wide range of images. The network has an image input size of 299-by-299, and the output is a list of estimated class probabilities.

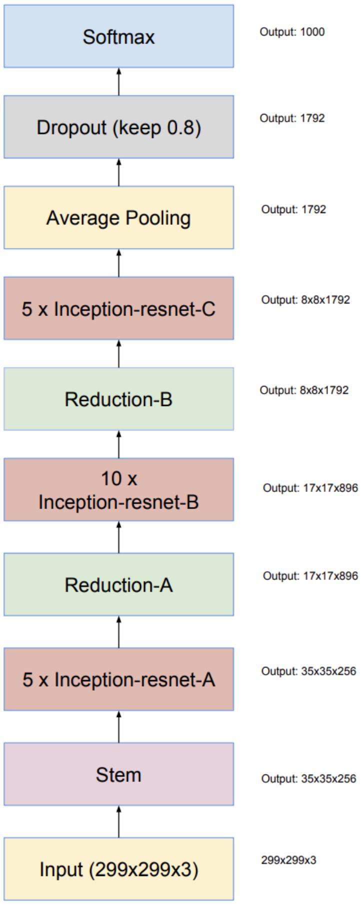


Figure 3: Schema for Inception RESNET v1 and v2 networks[6]

It is formulated based on a combination of the Inception structure and the Residual connection. In the Inception-Resnet block, multiple sized convolutional filters are combined with residual connections. The usage of residual connections not only avoids the degradation problem caused by deep structures but also reduces the training time.

Step 3: Downloading our own Dataset:

We used the OIDv6 dataset and extracted 11 subclasses which is specific to our own model. The 11 subclasses we extracted were Chainsaw, Hammer, Rifle, Knife, Dagger, Drill, Handgun, Shotgun, Sword, Tank and Weapon.

Open Images is a dataset of ~9M images annotated with image-level labels, object bounding boxes, object segmentation masks, visual relationships, and localized narratives:

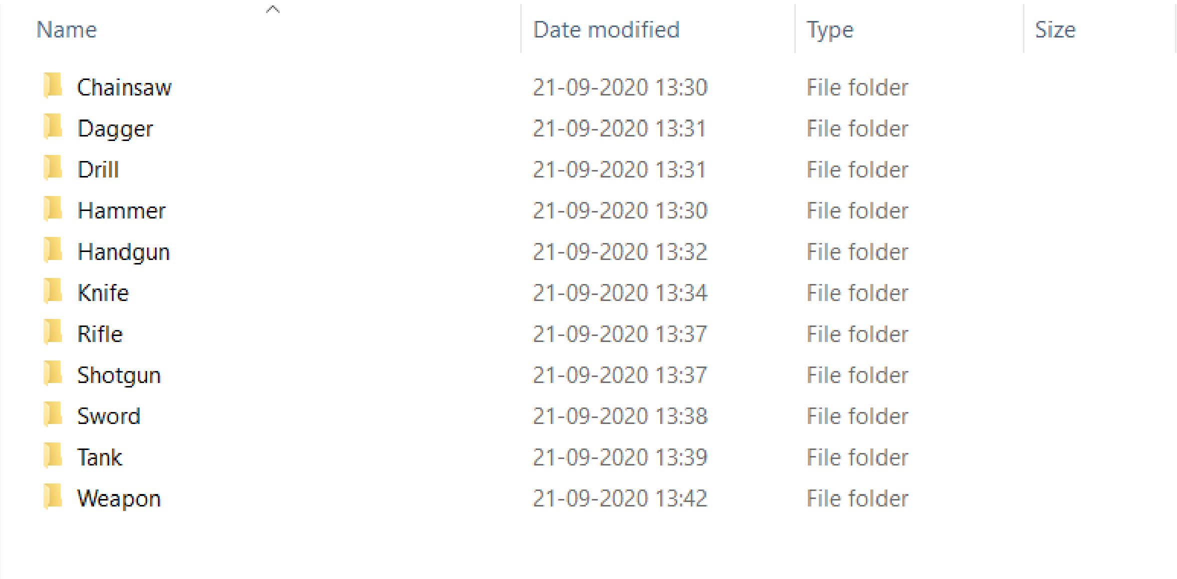
It contains a total of 16M bounding boxes for 600 object classes on 1.9M images, making it the largest existing dataset with object location annotations. The boxes have been largely manually drawn by professional annotators to ensure accuracy and consistency. The images are very diverse and often contain complex scenes with several objects (8.3 per image on average).

Open Images also offers visual relationship annotations, indicating pairs of objects in particular relations (e.g., "woman playing guitar", "beer on table"), object properties (e.g., "table is wooden"), and human actions (e.g., "woman is jumping"). In total it has 3.3M annotations from 1,466 distinct relationship triplets.

In V5 we added segmentation masks for 2.8M object instances in 350 classes. Segmentation masks mark the outline of objects, which characterizes their spatial extent to a much higher level of detail.

In V6 we added 675k localized narratives: multimodal descriptions of images consisting of synchronized voice, text, and mouse traces over the objects being described. (Note we originally launched localized narratives only on train in V6, but since July 2020 we also have validation and test covered.)

Finally, the dataset is annotated with 59.9M image-level labels spanning 19,957 classes. We believe that having a single dataset with unified annotations for image classification, object detection, visual relationship detection, instance segmentation, and multimodal image descriptions will enable to study these tasks jointly and stimulate progress towards genuine scene understanding.



Name	Date modified	Type	Size
Chainsaw	21-09-2020 13:30	File folder	
Dagger	21-09-2020 13:31	File folder	
Drill	21-09-2020 13:31	File folder	
Hammer	21-09-2020 13:30	File folder	
Handgun	21-09-2020 13:32	File folder	
Knife	21-09-2020 13:34	File folder	
Rifle	21-09-2020 13:37	File folder	
Shotgun	21-09-2020 13:37	File folder	
Sword	21-09-2020 13:38	File folder	
Tank	21-09-2020 13:39	File folder	
Weapon	21-09-2020 13:42	File folder	

Figure 4: 11 Subclasses of our OIDv6 dataset

Step 4: Transfer Learning:

Transfer learning is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task. It is a popular approach in deep learning where pre-trained models are used as the starting point on computer vision and natural language processing tasks given the vast compute and time resources required to develop neural network models on these problems and from the huge jumps in skill that they provide on related problems.

In transfer learning, we first train a base network on a base dataset and task, and then we repurpose the learned features, or transfer them, to a second target network to be trained on a target dataset and task. This process will tend to work if the features are general, meaning suitable to both base and target tasks, instead of specific to the base task.

1. Using this approach improving the layers of our model to be more accurate on our custom classes to increase its speed and accuracy.
2. We will apply transfer learning to our model so that the model becomes specific to our learning and we can produce desired outputs for a specific domain of dataset.

Transfer learning is the improvement of learning in a new task through the transfer of knowledge from a related task that has already been learned. While most machine learning algorithms are designed to address single tasks, the development of algorithms that facilitate transfer learning is a topic of ongoing interest in the machine-learning community.

Machine learning and data mining techniques have been used in numerous real-world applications. An assumption of traditional machine learning methodologies is the training data and testing data are taken from the same domain, such that the input feature space and data distribution characteristics are the same. However, in some real-world machine learning scenarios, this assumption does not hold. There are cases where training data is expensive or difficult to collect. Therefore, there is a need to create high-performance learners trained with more easily obtained data from different domains. This methodology is referred to as transfer learning.

There are different strategies and implementations for solving a transfer learning problem. The majority of the homogeneous transfer learning solutions employ one of three general strategies which include trying to correct for the marginal distribution difference in the source, trying to correct for the conditional distribution difference in the source, or trying to correct both the marginal and conditional distribution differences in the source. The majority of the heterogeneous transfer learning solutions are focused on aligning the input spaces of the source and target domains with the assumption that the domain distributions are the same. If the domain distributions are not equal, then further domain adaptation steps are needed. Another important aspect of a transfer learning solution is the form of information transfer (or what is being transferred). The form of information transfer is categorized into four general Transfer Categories.

The first Transfer Category is transfer learning through instances. A common method used in this case is for instances from the source domain to be reweighted in an attempt to correct for marginal distribution differences. These reweighted instances are then directly used in the target domain for training. These reweighting algorithms work best when the conditional distribution is the same in both domains.

The second Transfer Category is transfer learning through features. Feature-based transfer learning approaches are categorized in two ways. The first approach transforms the features of the source through reweighting to more closely match the target domain. This is referred to as asymmetric feature transformation. The second approach discovers underlying meaningful structures between the domains to find a common latent feature space that has predictive qualities while reducing the marginal distribution between the domains. This is referred to as symmetric feature transformation.

The third transfer category is to transfer knowledge through shared parameters of source and target domain learner models or by creating multiple source learner models and optimally combining the reweighted learners (ensemble learners) to form an improved target learner.

The last transfer category (and the least used approach) is to transfer knowledge based on some defined relationship between the source and target domains.

1. Transfer learning for NLP:

Textual data presents all sorts of challenges when it comes to ML and deep learning. These are usually transformed or vectorized using different techniques. Embeddings, such as Word2vec and FastText, have been prepared using different training datasets. These are utilized in different tasks, such as sentiment analysis and document classification, by transferring the knowledge from the source tasks. Besides this, newer models like the Universal Sentence Encoder and BERT definitely present a myriad of possibilities for the future.

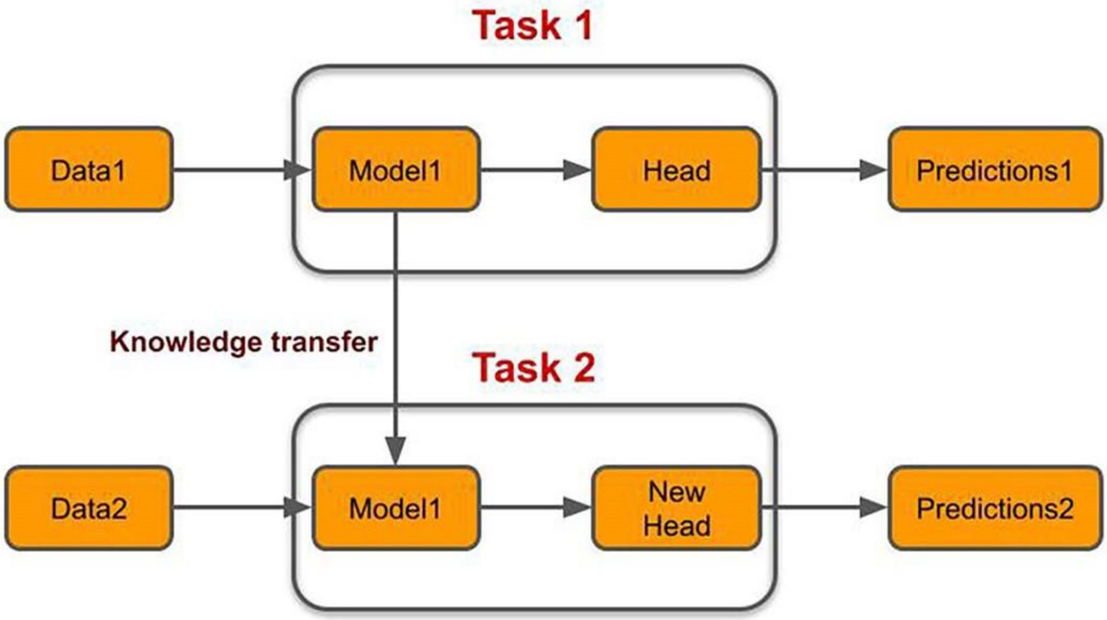


Figure 5: Transfer Learning for NLP[9]

2. Transfer Learning for Audio/Speech

Similar to domains like NLP and Computer Vision, deep learning has been successfully used for tasks based on audio data. For instance, Automatic Speech Recognition (ASR) models developed for English have been successfully used to improve speech recognition performance for other languages, such as German. Also, automated-speaker identification is another example where transfer learning has greatly helped.

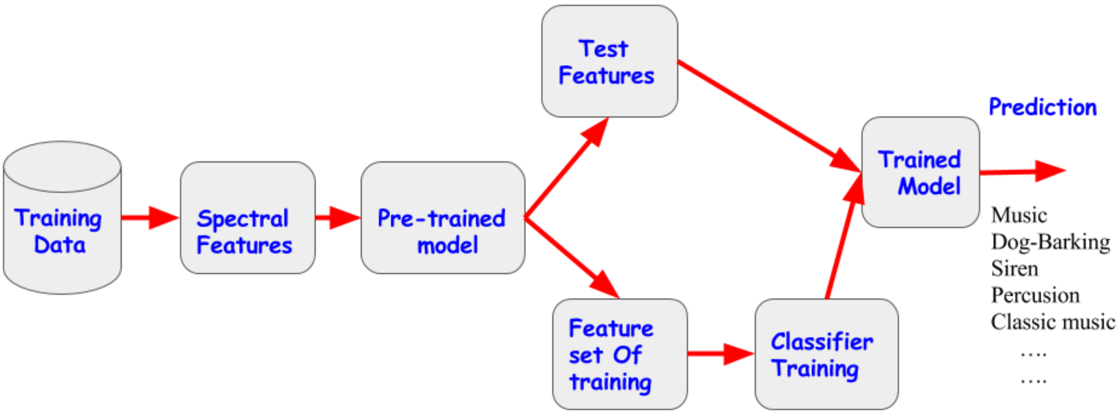


Figure 6: Audio classification using Transfer Learning Approach[3]

3. Transfer Learning for Object Detection

Deep convolutional neural network models may take days or even weeks to train on very large datasets.

A way to short-cut this process is to re-use the model weights from pre-trained models that were developed for standard computer vision benchmark datasets, such as the ImageNet image recognition tasks. Top performing models can be downloaded and used directly, or integrated into a new model for your own computer vision problems.

Transfer learning involves using models trained on one problem as a starting point on a related problem. Transfer learning is flexible, allowing the use of pre-trained models directly, as feature extraction preprocessing, and integrated into entirely new models. Keras provides convenient access to many top performing models on the ImageNet image recognition tasks such as VGG, Inception, and ResNet.

Transfer learning generally refers to a process where a model trained on one problem is used in some way on a second related problem. In deep learning, transfer learning is a technique whereby a neural network model is first trained on a problem similar to the problem that is being solved. One or more layers from the trained model are then used in a new model trained on the problem of interest.

This is typically understood in a supervised learning context, where the input is the same but the target may be of a different nature. For example, we may learn about one set of visual categories, such as cats and dogs, in the first setting, then learn about a different set of visual categories, such as ants and wasps, in the second setting.

Transfer learning has the benefit of decreasing the training time for a neural network model and can result in lower generalization error. The weights in re-used layers may be used as the starting point for the training process and adapted in response to the new problem. This usage treats transfer learning as a type of weight initialization scheme. This may be useful when the first related problem has a lot more labeled data than the problem of interest and the similarity in the structure of the problem may be useful in both contexts.

A range of high-performing models have been developed for image classification and demonstrated on the annual ImageNet Large Scale Visual Recognition Challenge, or ILSVRC.

This challenge, often referred to simply as ImageNet, given the source of the image used in the competition, has resulted in a number of innovations in the architecture and training of convolutional neural networks. In addition, many of the models used in the competitions have been released under a permissive license.

These models can be used as the basis for transfer learning in computer vision applications. This is desirable for a number of reasons, not least:

1. Useful Learned Features: The models have learned how to detect generic features from photographs, given that they were trained on more than 1,000,000 images for 1,000 categories.

2. State-of-the-Art Performance: The models achieved state of the art performance and remain effective on the specific image recognition task for which they were developed.

3. Easily Accessible: The model weights are provided as free downloadable files and many libraries provide convenient APIs to download and use the models directly. The model weights can be downloaded and used in the same model architecture using a range of different deep learning libraries, including Keras.

The use of a pre-trained model is limited only by your creativity. For example, a model may be downloaded and used as-is, such as embedded into an application and used to classify new photographs.

Alternately, models may be downloaded and use as feature extraction models. Here, the output of the model from a layer prior to the output layer of the model is used as input to a new classifier model.

Recall that convolutional layers closer to the input layer of the model learn low-level features such as lines, that layers in the middle of the layer learn complex abstract features that combine the lower level features extracted from the input, and layers closer to the output interpret the extracted features in the context of a classification task.

Armed with this understanding, a level of detail for feature extraction from an existing pre-trained model can be chosen. For example, if a new task is quite different from classifying objects in photographs (e.g. different to ImageNet), then perhaps the output of the pre-trained model after the few layers would be appropriate. If a new task is quite similar to the task of classifying objects in photographs, then perhaps the output from layers much deeper in the model can be used, or even the output of the fully connected layer prior to the output layer can be used.

The pre-trained model can be used as a separate feature extraction program, in which case input can be pre-processed by the model or portion of the model to a given an output (e.g., vector of numbers) for each input image, that can then use as input when training a new model.

Alternately, the pre-trained model or desired portion of the model can be integrated directly into a new neural network model. In this usage, the weights of the pre-trained can be frozen so that they are not updated as the new model is trained. Alternately, the weights may be updated during the training of the new model, perhaps with a lower learning rate, allowing the pre-trained model to act like a weight initialization scheme when training the new model.

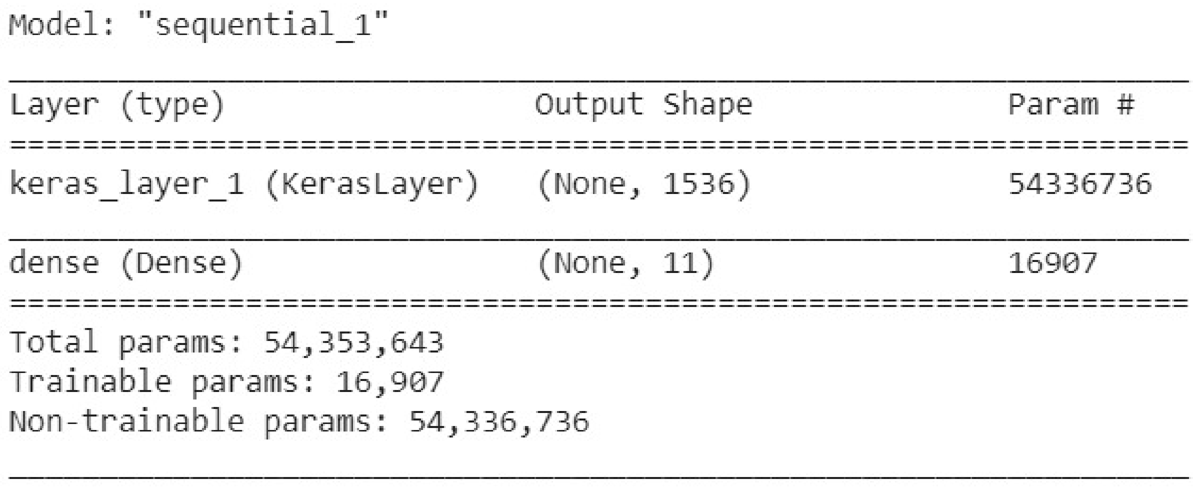
There are perhaps a dozen or more top-performing models for image recognition that can be downloaded and used as the basis for image recognition and related computer vision tasks.

Perhaps three of the more popular models are as follows: VGG (e.g., VGG16 or VGG19), Google Net (e.g. InceptionV3), Residual Network (e.g. ResNet50).
These models are both widely used for transfer learning both because of their performance, but also because they were examples that introduced specific architectural innovations, namely consistent and repeating structures (VGG), inception modules (Google Net), and residual modules (ResNet). Keras provides access to a number of top-performing pre-trained models that were developed for image recognition tasks.

They are available via the Applications API, and include functions to load a model with or without the pre-trained weights, and prepare data in a way that a given model may expect (e.g., scaling of size and pixel values).

Step 5: Training our Model:

TensorFlow Hub also distributes models without the top classification layer. So, we used headless model of inception_resnet_v2 to easily do transfer learning.
We then create the feature extractor. Using trainable=False to freeze the variables in the feature extractor layer, so that the training only modifies the new classifier layer.

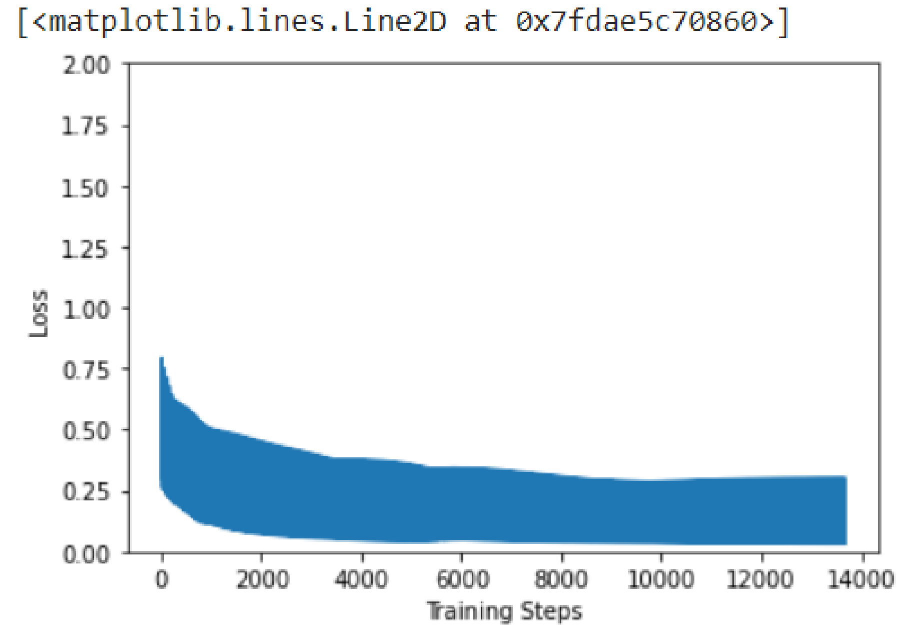


```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
keras_layer_1 (KerasLayer)   (None, 1536)              54336736
_________________________________________________________________
dense (Dense)                (None, 11)                16907
=================================================================
Total params: 54,353,643
Trainable params: 16,907
Non-trainable params: 54,336,736
_________________________________________________________________
```

Figure 7: Model Summary of our Model

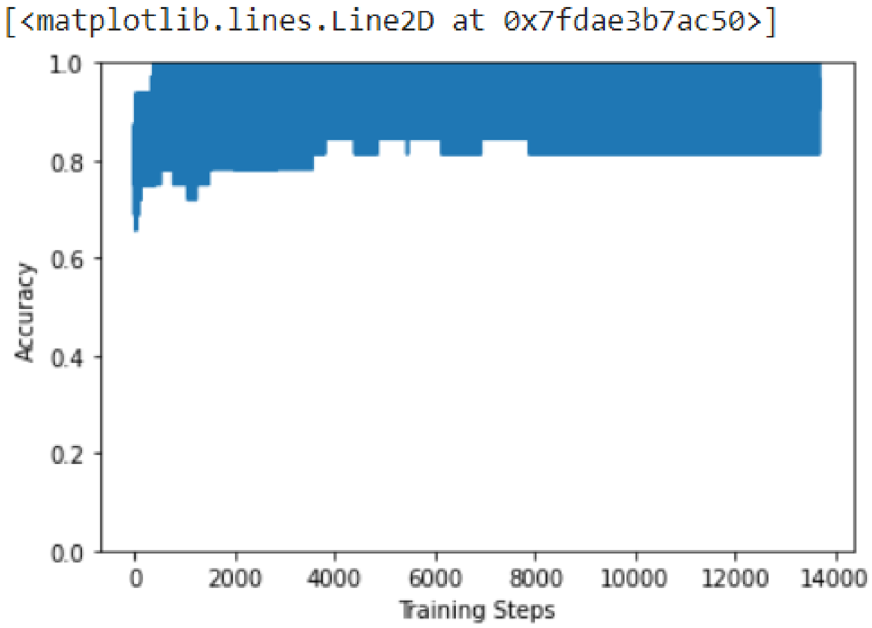
Step 6: Making Predictions:

After fitting our customized model, and training it with a number of epochs, we made predictions on different batches of images using our model. After training our model with at around 160 epochs, the accuracy of our model reached its convergence. We got an accuracy of approx. 91% which is a pretty good score for our model.



```
[<matplotlib.lines.Line2D at 0x7fdae5c70860>]
```

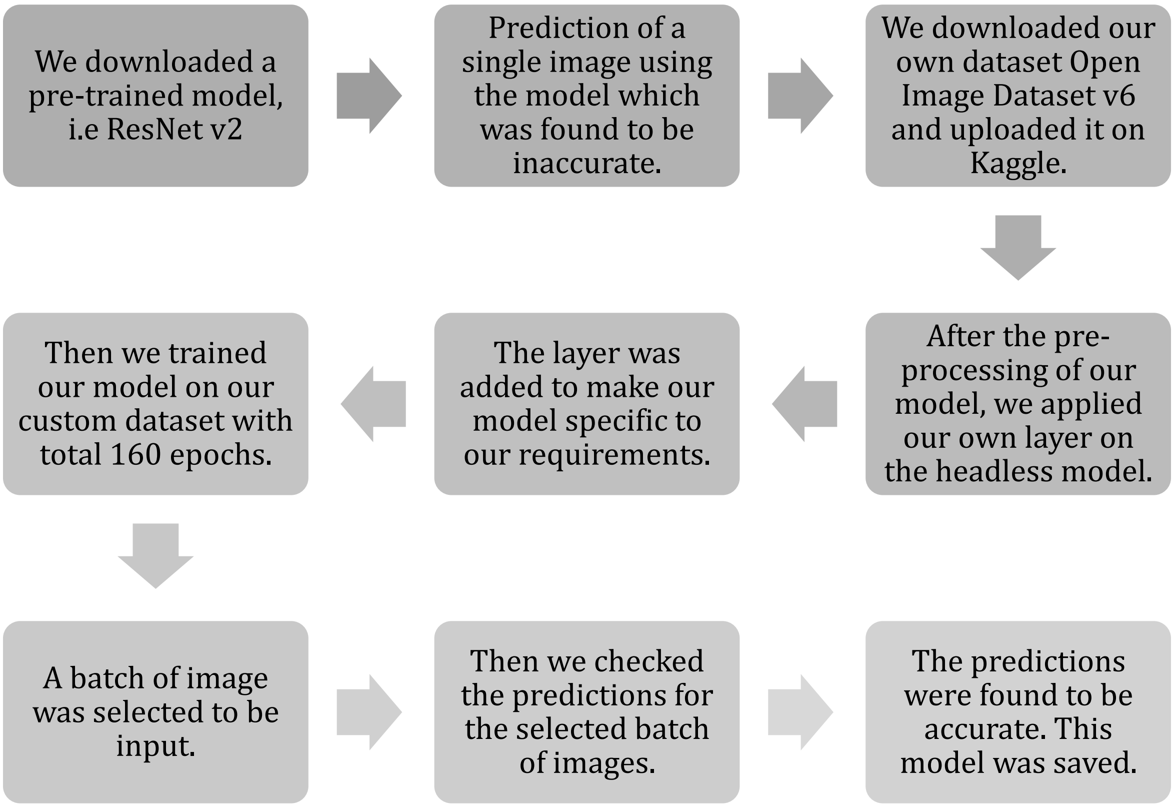
Figure 8 : Plot of Training Steps vs Loss



```
[<matplotlib.lines.Line2D at 0x7fdae3b7ac50>]
```

Figure 9 : Plot of Training Steps vs Accuracy

A basic summary of the workflow of our project is displayed below.



We downloaded a pre-trained model, i.e ResNet v2

Prediction of a single image using the model which was found to be inaccurate.

We downloaded our own dataset Open Image Dataset v6 and uploaded it on Kaggle.

After the pre-processing of our model, we applied our own layer on the headless model.

The layer was added to make our model specific to our requirements.

Then we trained our model on our custom dataset with total 160 epochs.

A batch of image was selected to be input.

Then we checked the predictions for the selected batch of images.

The predictions were found to be accurate. This model was saved.

Figure 10 : Basic Workflow of our Project

CHAPTER 3: RESULTS AND DISCUSSION

We first downloaded the pre trained classifier and run it on batch of 30 images to get the predictions as shown. Then we downloaded the headless version, i.e., without the top classification layer of the above and froze the variables in the feature extractor layer, so that the training only modifies the new classifier layer. After this we start training the model on our custom dataset with total 160 epochs. Now after, even just a few training iterations, we can already see that the model is making progress on the task. Now we run this trained model again on the 30 batch images to see the changes after achieving transfer learning.

Our transfer learning model reached an accuracy of approximately 91% when it was implemented using 160 epochs. We incurred a loss of 0.17.

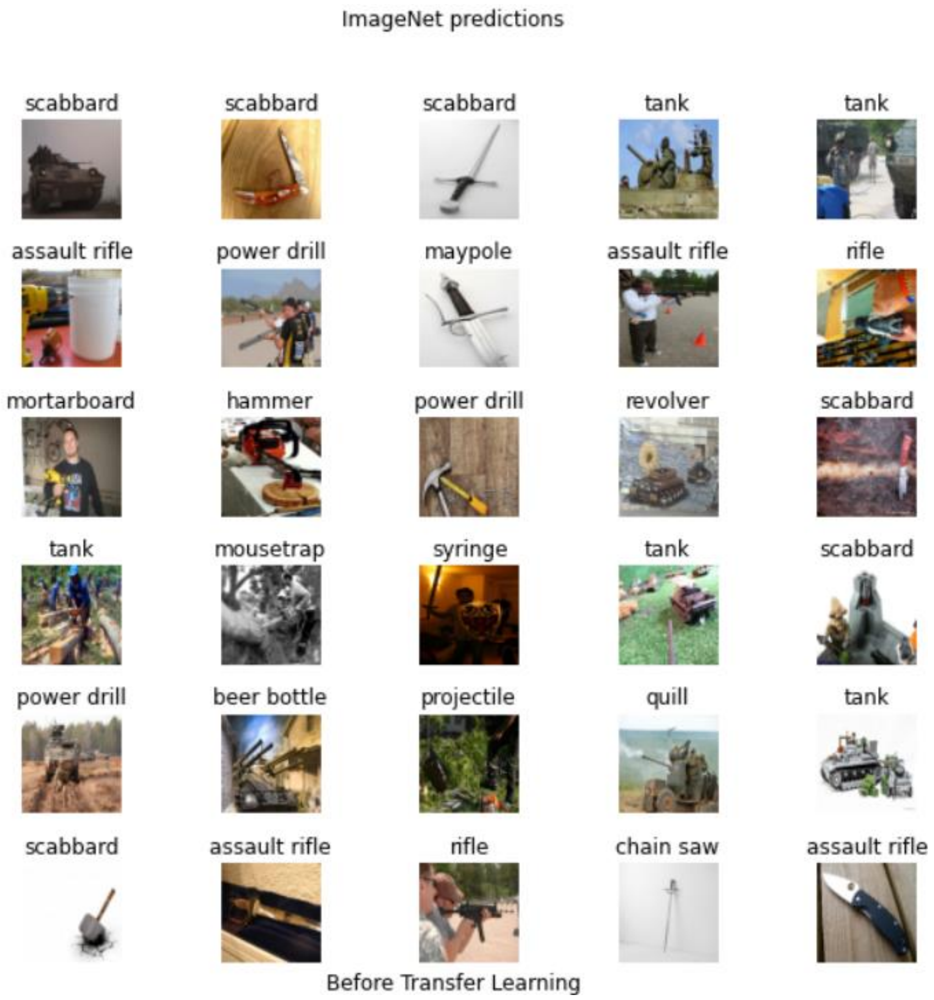


Figure 11: Predictions made by our model before Transfer Learning

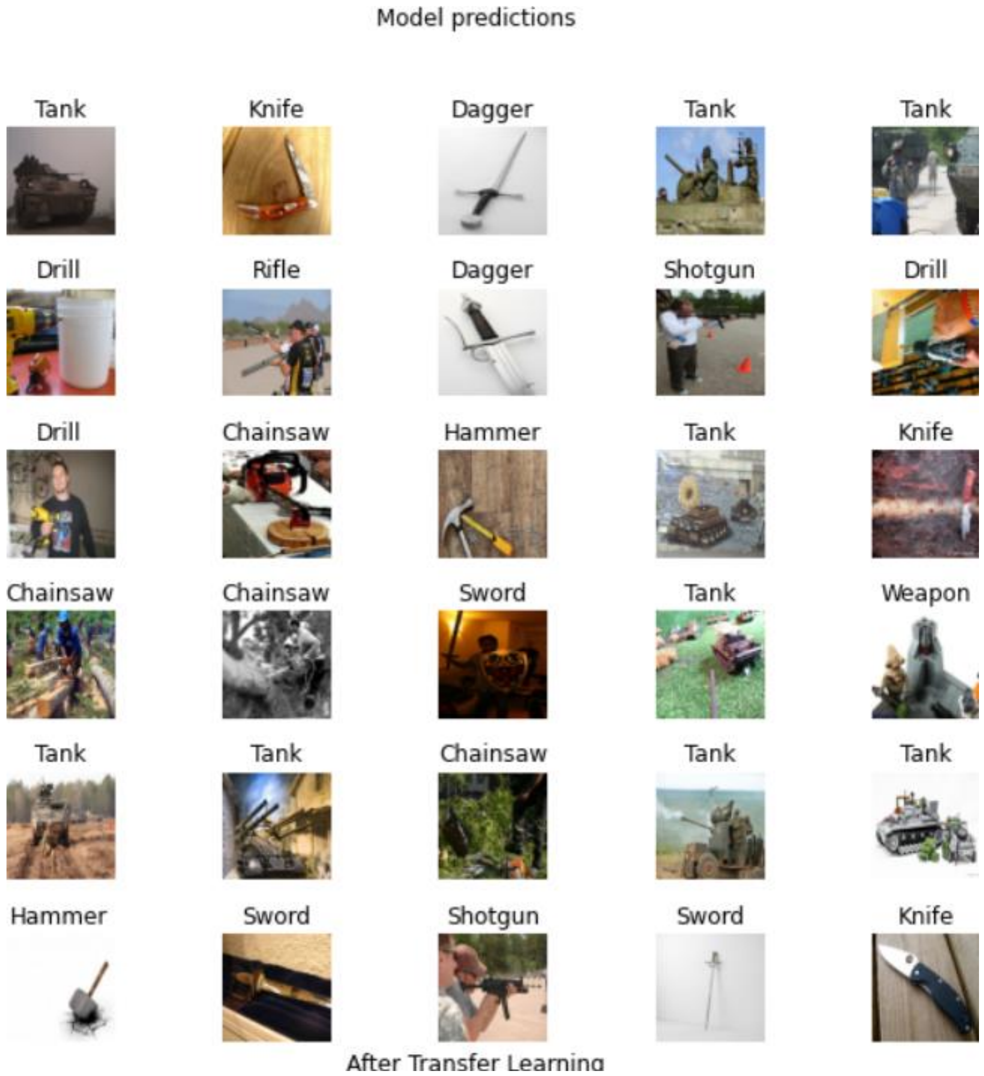


Figure 12: Predictions made by our model after Transfer Learning

CHAPTER 4: CONCLUSION

This paper gives detailed techniques which may be used in intruder detection or anomalous activity detection. When properly established, the proposed system could become an effective tool for discriminating genuine threats from false alarms in a practical situation.

Digital image processing techniques can be used for the analysis of CCTV security camera images to successfully provide automatic detection of persons, and alleviate the work overhead of a human operator. The application of such techniques eliminates the need for constant operator supervision in an area to be kept secure.

We plan to continue our work on our code in order to provide a complete and ready-to-market solution for CCTV operators. We intend to apply an OpenCV framework which could take in live footage as its input. We'd also try to inculcate a better variety of classes for our dataset in order for it to work in a real-world scenario. Another thing which could be done is to apply or tweak more RESNET layers to make the model even more specific to our requirements. Another research direction that we will pursue is the introduction of new modalities: the introduction of pan-tilt-zoom cameras, the infrared spectrum for low light conditions and thermography for better distinction of the dangerous tool from the background. We also foresee extending the number of detected classes by other firearm types and by other dangerous objects (e.g., machetes, clubs and bats).

REFERENCES

[1] Shahid Karim, Ye Zhang, and Asif Laghari. Image processing based proposed drone for detecting and controlling street crimes. In 17th IEEE International Conference on Communication Technology, 10 2017.

[2] Chun Che Fung and Nicholas Jerrat. A neural network based intelligent intruders detection and tracking system using cctv images. In 2000 TENCON Proceedings. Intelligent Systems and Technologies for the New Millennium (Cat. No. 00CH37119), volume 2, pages 409–414. IEEE, 2000.

[3] Micha l Grega, Andrzej Matiola´nski, Piotr Guzik, and Miko laj Leszczuk. Automated detection of firearms and knives in a cctv image. Sensors, 16(1):47, 2016.

[4] Kamran Ali Changezi and Muhammad Wasil Zafar. An object detection using image processing in digital forensics science. Journal of Independent Studies and Research-Computing, 16(1):1–10, 2018.

[5] Chun-Ku Lee, Meng-Fen Ho, Wu-Sheng Wen, and Chung-Lin Huang. Abnormal event detection in video using n-cut clustering. In 2006 International Conference on Intelligent Information Hiding and Multimedia, pages 407–410. IEEE, 2006.

[6] Rutvik Kakadiya, Reuel Lemos, Sebin Mangalan, Meghna Pillai, and Sneha Nikam. Ai based automatic robbery/theft detection using smart surveillance in banks. In 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), pages 201–204. IEEE, 2019.

[7] JA Freer, BJ Beggs, HL Fernandez-Canque, F Chevrier, and A Goryashko. Automatic video surveillance with intelligent scene monitoring and intruder detection. In 1996 30th Annual International Carnahan Conference on Security Technology, pages 89–94. IEEE, 1996.

[8] Tomasz Kryjak, Mateusz Komorkiewicz, and Marek Gorgon. Real-time moving object detection for video surveillance system in fpga. In Proceedings of the 2011 Conference on Design & Architectures for Signal & Image Processing (DASIP), pages 1–8. IEEE, 2011.

[9] Nawin Kongurgsa, Narumol Chumuang, and Mahasak Ketcham. Real-time intrusion—detecting and alert system by image processing techniques. In 2017 10th International Conference on Ubi-media Computing and Workshops (Ubi-Media), pages 1–6. IEEE, 2017.

[10] Roberto Olmos, Siham Tabik, and Francisco Herrera. Automatic handgun detection alarm in videos using deep learning. Neurocomputing, 275:66–72, 2018.

[11] Zhenfeng Shao, Jiajun Cai, and Zhongyuan Wang. Smart monitoring cameras driven intelligent processing to big surveillance video data. IEEE Transactions on Big Data, 4(1):105–116, 2017.

[12] Rainer Lienhart, Luhong Liang, and Alexander Kuranov. A detector tree of boosted classifiers for real-time object detection and tracking. In 2003 International Conference on Multimedia and Expo. ICME’03. Proceedings (Cat. No. 03TH8698), volume 2, pages II–277. IEEE, 2003.