AN IOT BASED SYSTEM FOR CHILD SAFETY

Kushagra Vyas[1], Naman Agrawal[2], Devashish Shukla[3], Amisha Bharti[4],

Ms.Prathima Mabel[5]

[1,2,3,5]B.E. Student, Dept. of Information Science and Engineering, Dayananda Sagar College of Engineering, Kumaraswamy Layout, Bengaluru, India
[5]Assistant Professor, Dept. of Information Science and Engineering, Dayananda Sagar College of Engineering, Kumaraswamy Layout, Bengaluru, India

---***---

ABSTRACT:

For parents the safety of their child is a big question. When there are so many bad influence outside and the crime rate against minors is so high parents who cannot remain with their child for the entire day, need something to-monitor them. This something can be provided to them in the form of an IOT based smart band. Sensors attached with this band can help to detect the presence of any bad influence, unwanted person or a threat around the child. If an issue is detected by the sensors, an alert containing the GPS location of the child and live feed of his environment will be sent to an app on parent's phone via internet(GSM module). Smoke sensor and pulse sensor attached with the device can help to detect any abnormal situation around the child. Usage of image processing helps to analyze the situation and to classify it as dangerous or non-dangerous.

Keywords *: IOT , GSM , Image Processing.*

1. INTRODUCTION:

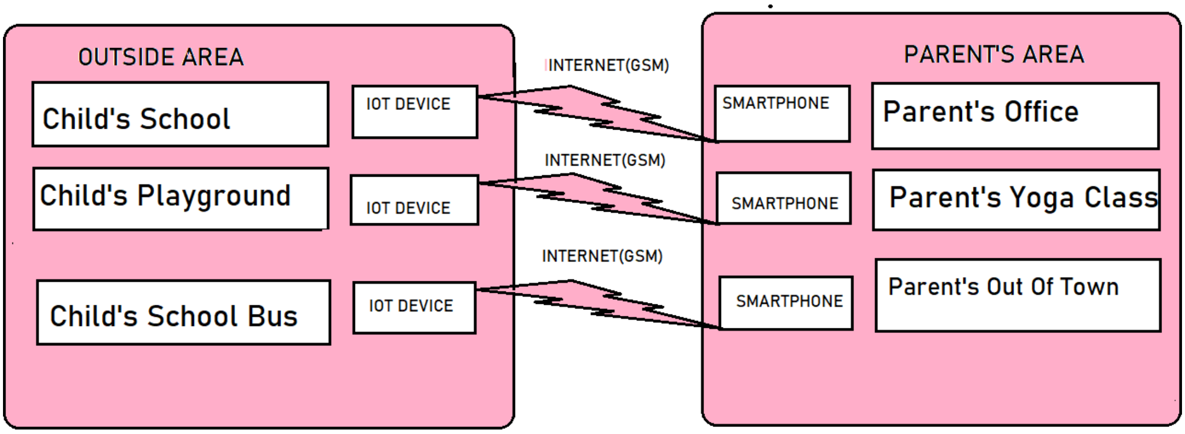
Parent are happy when their child is safe and happy. They take every step to ensure that their child is happy. But children are most vulnerable when they are outside of the protective hemisphere of parents.

When a child goes to his school, music-class and playground. These are the times when he is most vulnerable to crimes and bad influence. With a device that can monitor a child's environment this issue can be resolved . The proposed system provides a combination of technologies like Internet Of Things, Image Processing and Global Positioning System. This system uses sensors and Pi-camera to detect any unwanted influence around the children. Pulse Rate sensor detects any threatening and unwanted situation around the child, whereas the Pi-camera takes in the live feed of the child's environment, performs image processing over it and in case of a danger passes an alert message to an application running on parent's smartphone, the alert message contains the GPS location of the child taken at an interval of 5 to 15 minutes and the live feed(video) of the his environment. All these data taken with the help of a GSM module and a pi-camera helps the parent to track the condition and location of their child once he has fallen into a trouble.

Apart from providing an automated mechanism to predict and transmit the presence of a danger to parent, the proposed system also provides a push button, which enables the child to manually raise an alert in case the sensors or the camera don't work properly. The alert message described above is then forwarded to the parent.

- **IOT in Human Safety** Modern world, owing to the increased rate in crimes require an efficient human being monitoring system that can safeguard the well-being of a person, by allowing another to perform health and safety monitoring.
- Internet of Things makes this process easy. A similar project when developed for a child should take into consideration the presence of a manual alternative in case automation fails

- **Model Of The Problem**:



- **Purpose** : With the help of proposed system, parents can monitor the environment of the child and ensure their ward' safety without much human intervention.

2. PROBLEM STATEMENT

In today's society where crime has increased beyond the limits children are the most vulnerable target for various criminal activities. For parent's their ward's safety is the most important issue but they cannot stay with their children throughout the day.

When children are away from their parent's, during this time they are most vulnerable to crimes and bad influence.

Children are the future of the society, in their safety and development lies the development of the society. Therefore their safety in no case should be compromised. This safety issue can be easily resolved with the help of the proposed system.

3. LITERATURE SURVEY

The advancements in the field of Internet of Things has lead to development of many significant applications in various fields and industries .One of them is remote monitoring.

As it is known to everyone parent don't get sufficient time to spend with their children and it is not possible for them to remain with their children all time. These durations when children are away from their parent, children are most vulnerable to crimes and bad company .These things are really harmful and dangerous for children.

Bad influence around a child may involve a smoker, unwanted physical touches and abuse .A child on his way to school may be kidnapped or may encounter some other danger .Although a child can be protected from such incidents by not letting him out alone but this is not possible for every parent .Due to an imbalance between the number of citizens and cops, cops may not be able to protect every one, every time. According to a study done in 2004 an aggregate of 5996 children went missing, out of which 4092 children were found by police but 1904 of them are still missing .Usage of an appropriate device that detects any danger around the child and notifies parent in case of a danger can solve this issue.

Smart phone based apps, location detector and other alert devices are there but the limitations with such devices are that they have to be activated manually by the user(children), as an example in 2017, "Smart Security Solution for Women and Children Safety Based on GPS Using IoT" was proposed, but this device sends the location of the victim only after he/she presses an alert button which might be not possible in the case of an emergency. Also to examine the situation around the victim, some sort of camera capture is required.

This project comprises of a safety band for a child. It makes use of Image processing and sensors which detects the presence of any danger around the child. In case of danger an alert message along with the GPS location and a camera feed (current) of the child is sent to the parent. The projected system aims at identifying threats and bad influence around the child (even if he/she itself is not aware) and raising an alert automatically. Although it aims to provide an automatic action(alert) in case of a threat, a manual button to raise an alert is also provided.

Usage of Image Processing allows to detect any danger around the child (smoke, stick and knife) and immediately raise an alert to the parent.

Usage of smoke sensor detects presence of a smoke-related danger (example- fire).
pulse sensor helps to detect any stressful situation with the child. Combination of sensors and Image Processing allows for an automated detection of danger around the child.

4. OBJECTIVES

The objectives of this project are given as:

- To monitor a child, detect any danger(smoke, stick, knife) around him/her, and raise an alert (containing GPS location and camera feeds) to parent.
- To check and confirm that the child is not under any bad influence.
- To enable parent trace and monitor their children in case of an emergency (with the help of GPS module and pi camera.

5. PROPOSED SYSTEM

The system scans and detects any unusual situation around child. Classifies them as dangerous and non-dangerous and raises an alert to parent in case of a danger. This system makes sure that parent are always of their child's situation (in case he is away from them)

Advantages:

1. Ensures security of the child.
2. Enables parent to easily and efficiently monitor child.

6. METHODOLOGY

The projected device implements technologies like Image Processing and Internet Of Things into a safety band. These safety band detects any dangerous situation around the child (user) and raises an alert to the parent, thereby providing them a mean to monitor their child.

The system uses a Raspberry pi along with a pulse sensor to detect any stressful situation the child may be facing and a smoke sensor to detect any fire related accident that the child may be caught in.

Pi-camera attached with the device captures the surrounding of the child(video) performs image processing over it and checks for any dangerous object (stick, knife and smoke), presence of a dangerous object around the child may represent an actual danger.

Either if the sensors gets readings above a certain threshold or pi-camera detects any three of the above mentioned objects or the button is pressed(manually), an alert message containing the GPS location of the child along with the video is send to an app over parent's smartphone. The parent's can then decide whether their child is a danger or not. In case of a danger parent's can get GPS location of the child and a recording of his surrounding at a regular interval. This will help the parent track their child once he is caught in a dangerous situation (say kidnapped).

Presence of a burning cigarette around child can be detected by parent, by careful observation of the video (containing smoke) received on their app.

To maintain the efficiency of the device, the recoding by the pi-camera and sensors should be continuously done, so as to process them. All the videos containing stick, smoke and knife will be maintained over the server while rest of the video feeds will be deleted.

6.1 System Architecture

The system consists of three parts :

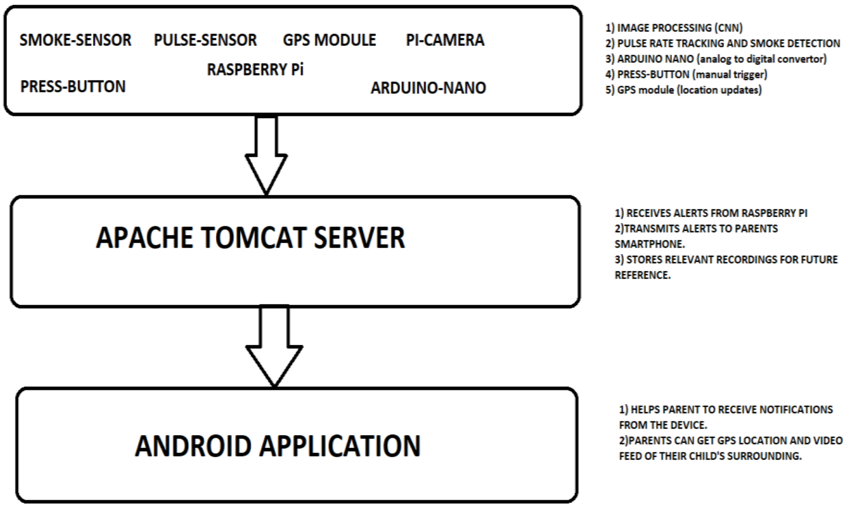
1. Hardware part / Application part
2. Middle Layer (Server)
3. Mobile Application (Parent's end)

1. Hardware part: This part interacts directly with the child's environment. It is responsible for taking the readings through sensors and capturing video from pi-camera, processing them and classifying them as danger symbols or not.

   Image Processing is performed over the video captured, by taking individual frames from it.

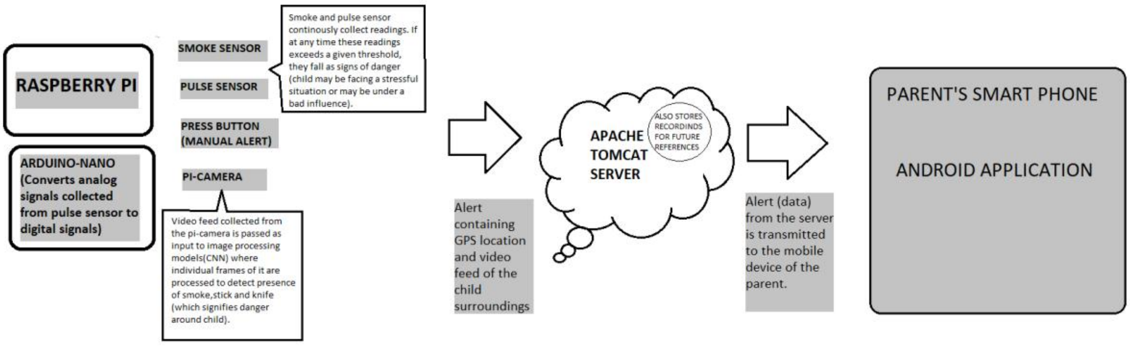
   Once danger is detected (either from Image Processing or readings taken in through sensors or by the manual press of alert) an alert containing GPS location of the child and video is send to the mobile application (parent).

   Hardware components used are Raspberry Pi 3, GPS module, Pi-Camera, Button, smoke sensor, pulse sensor, Arduino Nano (analog to digital convertor)
2. Middle Layer (Server): This layer is responsible for relaying the alerts(data) from the raspberry pi to the Mobile application(parent), it also stores the videos recordings taken in by the pi-camera that are of relevance (contains smoke, knife or stick). Server used for this application are Apache Tomcat.
3. Mobile Applications: Parents will receive the alert on an android application. Once they receive the alert containing location and video capture they can decide (by observing the video) whether an actual danger is there or not. After that they can receive GPS location and a video feed from the band at a regular interval.



6.2 Sequence Diagram

1) A danger is detected if any of the following is true.
   - A manual alert is raised by pressing the button.
   - Smoke sensor or heart beat sensor reads values above a threshold.
   - Smoke, knife or stick is detected in the video by means of Image Processing algorithms (CNN).
2) If a danger is detected in the previous step, an alert containing the GPS location of the child and a video feed of his surroundings are send to the apache tomcat server.
3) From the server, these alerts are transmitted to the parent's android device over an app.
4) Parents can then check the video and make out whether child is an actual danger?
5) In case of a danger, parents can get the GPS location and video feed of the child over a fixed interval.



IMPLEMENTATION

Step1: Convolutional Neural Networks:

Convolutional Neural Networks have a different architecture than regular Neural Networks. Regular Neural Networks transform an input by putting it through a series of hidden layers. Every layer is made up of a set of neurons, where each layer is fully connected to all neurons in the layer before. Finally, there is a last fully-connected layer — the output layer — that represent the predictions.

Convolutional Neural Networks are a bit different. First of all, the layers are organised in 3 dimensions: width, height and depth. Further, the neurons in one layer do not connect to all the neurons in the next layer but only to a small region of it. Lastly, the final output will be reduced to a single vector of probability scores, organized along the depth dimension.

CNN is composed of two major parts:

- **Feature Extraction**:
  In this part, the network will perform a series of convolutions and pooling operations during which the features are detected. If you had a picture of a zebra, this is the part where the network would recognize its stripes, two ears, and four legs.

- **Classification**:
  Here, the fully connected layers will serve as a classifier on top of these extracted features. They will assign a probability for the object on the image being what the algorithm predicts it is.

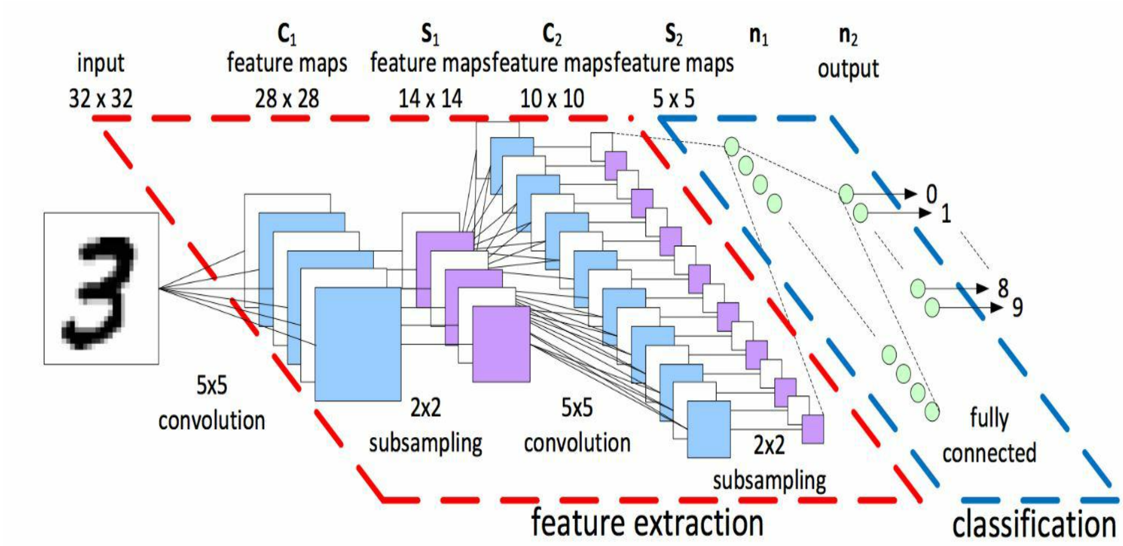


Fig1: Convolutional Neural Networks architecture

There are squares and lines inside the red dotted region which we will break it down later. The green circles inside the blue dotted region named classification is the neural network or multi-layer perceptron which acts as a classifier. The inputs to this network come from the preceding part named feature extraction.

Feature extraction is the part of CNN architecture from where this network derives its name. Convolution is the mathematical operation which is central to the efficacy of this algorithm. Lets understand on a high level what happens inside the red enclosed region. The input to the red region is the image which we want to classify and the output is a set of features. Think of features as attributes of the image, for instance, an image of a cat might have features like whiskers, two ears, four legs etc. A handwritten digit image might have features as horizontal and vertical lines or loops and curves. Later we'll see how do we extract such features from the image.

Step2: Feature Extraction: Convolution:

Convolution in CNN is performed on an input image using a filter or a kernel. To understand filtering and convolution you will have to scan the screen starting from top left to right and moving down a bit after covering the width of the screen and repeating the same process until you are done scanning the whole screen. For instance if the input image and the filter look like following:

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

Input

1	0	1
0	1	0
1	0	1

Filter / Kernel

Fig2: Input image and the filter

The filter (green) slides over the input image (blue) one pixel at a time starting from the top left. The filter multiplies its own values with the overlapping values of the image while sliding over it and adds all of them up to output a single value for each overlap until the entire image is traversed:

1	1	1	0	0
0	1	1	1	0
0	0	1x1	1x0	1x1
0	0	1x0	1x1	0x0
0	1	1x1	0x0	0x1

4	3	4
2	4	3
2	3	4

Fig3: Filter (green) slides over the input image

In the above Fig3 the value 4 (top left) in the output matrix (red) corresponds to the filter overlap on the top left of the image which is computed as:

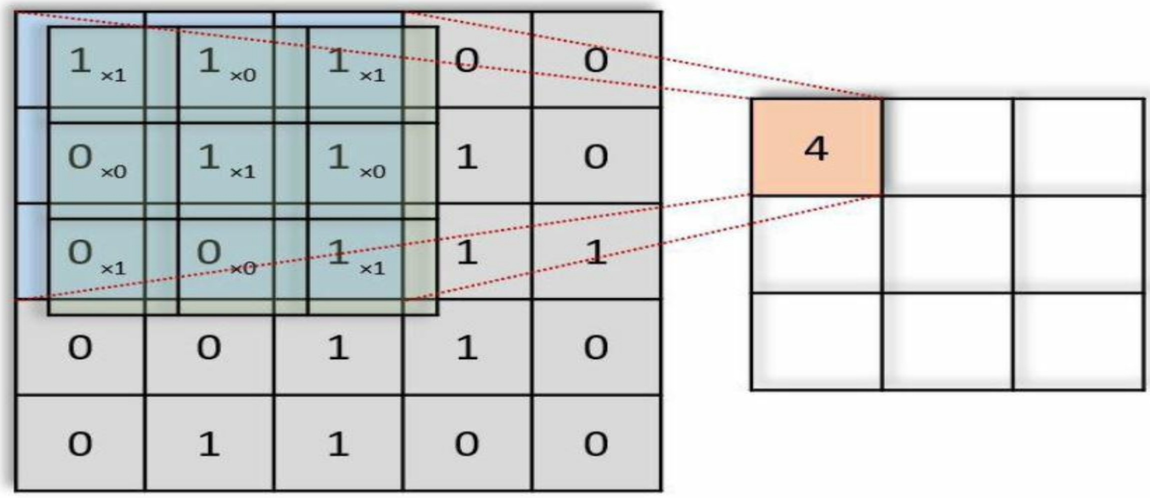


Fig4: 1st step of convolution

(1×1+0×1+1×1)+(0×0+1×1+1×0)+(1×0+0×0+1×1)=4

Similarly we compute the other values of the output matrix. Note that the top left value, which is 4, in the output matrix depends only on the 9 values (3x3) on the top left of the original image matrix. It does not change even if the rest of the values in the image change. This is the receptive field of this output value or neuron in our CNN. Each value in our output matrix is sensitive to only a particular region in our original image.

In the case of images with multiple channels (e.g. RGB), the Kernel has the same depth as that of the input image. Matrix Multiplication is performed between Kn and In stack ([K1,1],[K2,I2],[K3,I3]) and all the results are summed with the bias to give us a squashed one-depth channel Convoluted Feature Output:

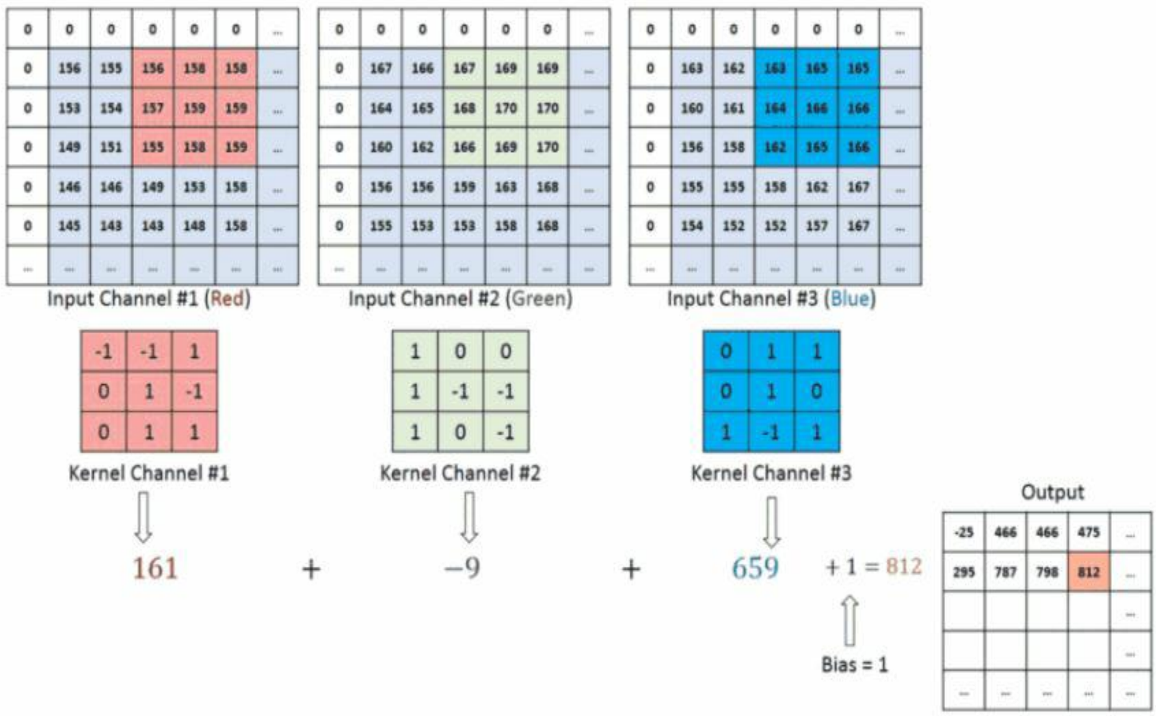


Fig5: Squashed one-depth channel convolution feature

Each neuron in the output matrix has overlapping receptive fields. The Fig6 below will give you a better sense of what happens in convolution.

Conventionally, the first Convolutional Layer is responsible for capturing the Low-Level features such as edges, color, gradient orientation, etc.

With added layers, the architecture adapts to the High-Level features as well, giving us a network which has the wholesome understanding of images in the data-set, similar to how we would.

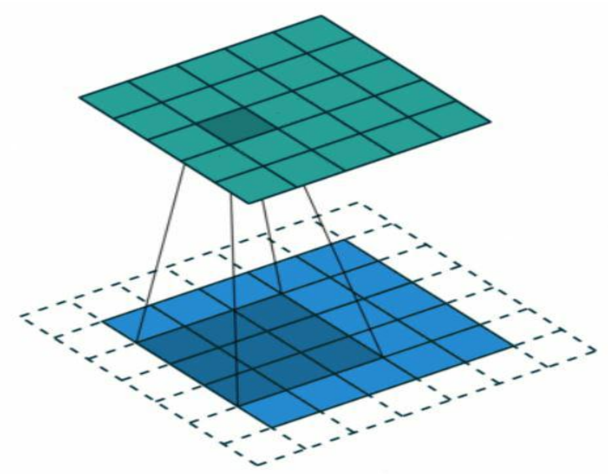


Fig6: Convolution example

Step3: Feature Extraction: padding:

There are two types of results to the operation — one in which the convoluted feature is reduced in dimensionality as compared to the input, and the other in which the dimensionality is either increased or remains the same. This is done by applying Valid Padding or Same Padding in the case of the latter. In above example our padding is 1.

In our example when we augment the 5x5x1 image into a 7x7x1 image and then apply the 3x3x1 kernel over it, we find that the convoluted matrix turns out to be of dimensions 5x5x1. It means our output image is with same dimensions as our output image (Same Padding).

On the other hand, if we perform the same operation without padding, in the output we'll receive an image with reduced dimensions. So our (5x5x1) image will become (3x3x1).

- **Feature Extraction: example:**

Let's say we have a handwritten digit image like the one below. We want to extract out only the horizontal edges or lines from the image. We will use a filter or kernel which when convoluted with the original image dims out all those areas which do not have horizontal edges:

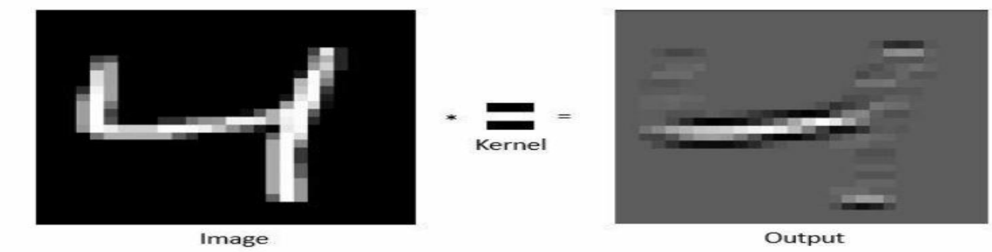


Fig7: Horizontal filter example

Notice how the output image only has the horizontal white line and rest of the image is dimmed. The kernel here is like a peephole which is a horizontal slit. Similarly for a vertical edge extractor the filter is like a vertical slit peephole and the output would look like:

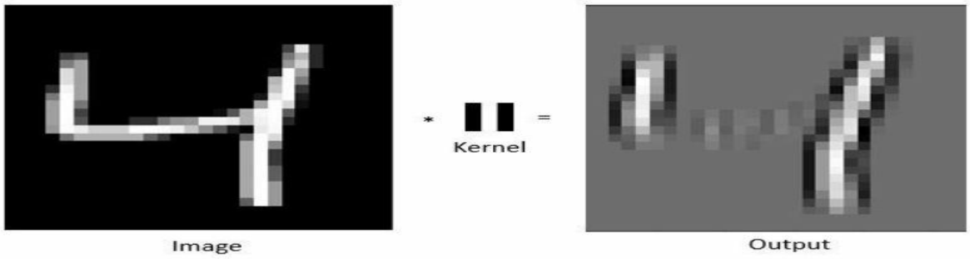


Fig8: Vertical filter example

Step4: Feature Extraction: Non-Linearity:

After sliding our filter over the original image the output which we get is passed through another mathematical function which is called an activation function. The activation function usually used in most cases in CNN feature extraction is ReLu which stands for Rectified Linear Unit. Which simply converts all of the negative values to 0 and keeps the positive values the same:

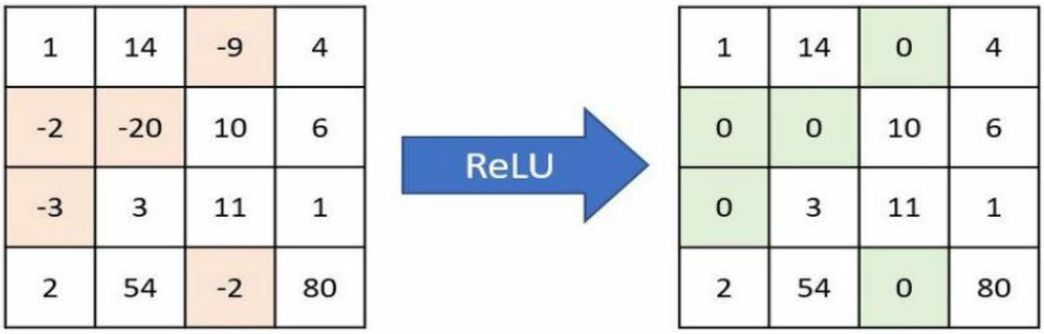


Fig9: CNN feature extraction with ReLu

After passing the outputs through ReLu functions they look like:

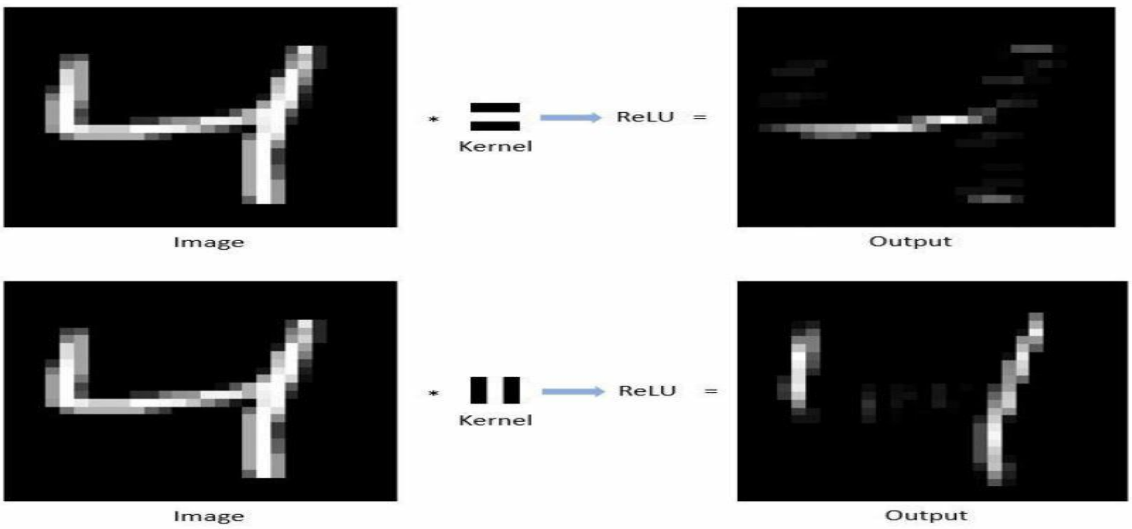


Fig10: Input image after filters with ReLu example

So for a single image by convolving it with multiple filters we can get multiple output images. For the handwritten digit here we applied a horizontal edge extractor and a vertical edge extractor and got two output images. We can apply several other filters to generate more such outputs images which are also referred as feature maps.

Step5: Feature Extraction: Pooling:

After a convolution layer once you get the feature maps, it is common to add a pooling or a sub-sampling layer in CNN layers. Similar to the Convolutional Layer, the Pooling layer is responsible for reducing the spatial size of the Convolved Feature. This is to decrease the computational power required to process the data through dimensionality reduction. Furthermore, it is useful for extracting dominant features which are rotational and positional invariant, thus maintaining the process of effectively training of the model. Pooling shortens the training time and controls over-fitting.

6.3 Two Types of Pooling:

Max Pooling and Average Pooling. Max Pooling returns the maximum value from the portion of the image covered by the Kernel. Max Pooling also performs as a Noise Suppressant. It discards the noisy activation altogether and also performs de-noising along with dimensionality reduction.

Average Pooling returns the average of all the values from the portion of the image covered by the Kernel. Average Pooling simply performs dimensionality reduction as a noise suppressing mechanism. Hence, we can say that Max Pooling performs a lot better than Average Pooling.

The Convolutional Layer and the Pooling Layer, together form the i-th layer of a Convolutional Neural Network. Depending on the complexities in the images, the number of such layers may be increased for capturing low-levels details even further, but at the cost of more computational power.

After going through the above process, we have successfully enabled the model to understand the features. Moving on, we are going to flatten the final output and feed it to a regular Neural Network for classification purposes.

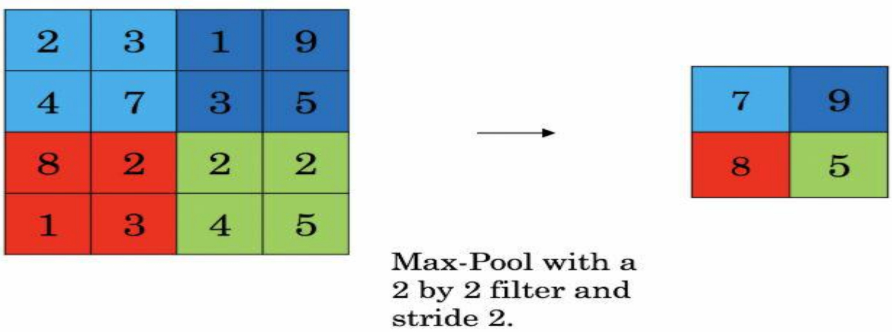


Fig11: Max Pooling example

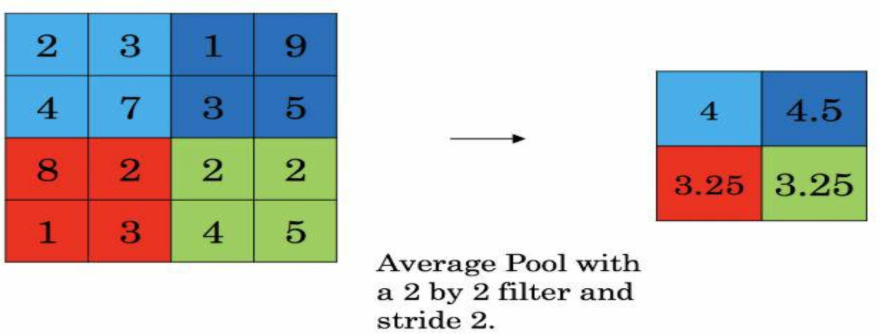


Fig12: Average Pooling example

Step6: Classification — Fully Connected Layer (FC Layer):

Adding a Fully-Connected layer is a (usually) cheap way of learning non-linear combinations of the high-level features as represented by the output of the convolutional layer. The Fully-Connected layer is learning a possibly non-linear function in that space. Example of CNN network:

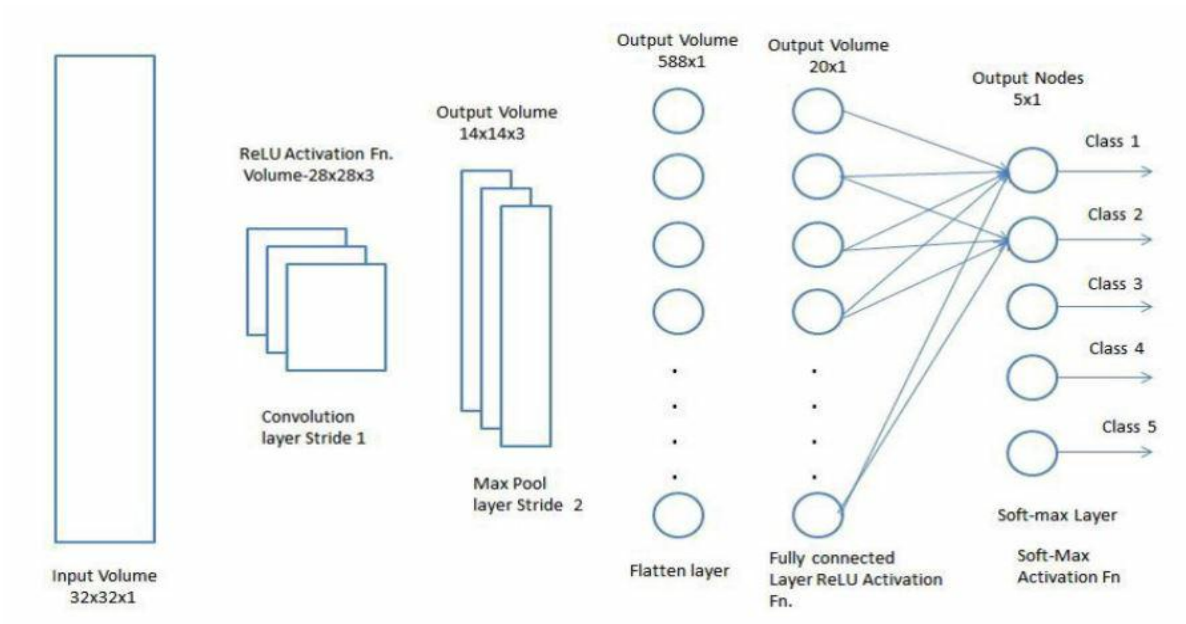


Fig13: Fully Connected model

Now that we have converted our input image into a suitable form, we shall flatten the image into a column vector. The flattened output is fed to a feed-forward neural network and backpropagation applied to every iteration of training. Over a series of epochs, the model is able to distinguish between dominating and certain low-level features in images and classify them using the Softmax Classification technique.

So now we have all the pieces required to build a CNN. Convolution, ReLU and Pooling. The output of max pooling is fed into the classifier which is usually a multi-layer perceptron layer. Usually in CNNs these layers are used more than once i.e. Convolution -> ReLU -> Max-Pool -> Convolution -> ReLU -> Max-Pool and so on.

7.0 PROJECT OUTCOMES

This project results in a device that can be used by parents to monitor their children when they are away. It helps the parent detect any danger around their child and take suitable action. If child goes missing the device can help to track him. This smart safety therefore allows the parent to ensure their child's safety while they are away. As this device is built on Image Processing and Internet of Things, it provides an automatics monitoring and alert raising system requiring little human intervention.

8.0 CONCLUSIONS

This project uses Image Processing and Internet of Things to built a smart band that can ensure the safety of a child while he is away from the parent. This way a child can be safe, and crimes against children can be reduced in a long run.

9.0 REFERENCES

1) Child Safety Wearable Devices With IOT by Ms. S.Sorna Valli and Mrs. A Jasmine Sugil - International Journal of Advanced Research in Basic Engineering Sciences and Technology (IJARBEST)
2) Child Safety Wearable Device by Mr. Akash Moodbidri and Mr. Hamid Shahnasser
3) GPS & GSM Based Child Tracking System Using Smartphone by Mr Al-Mazloum and Mr. E Omer
4) A Hybrid Model on Child Security and Activities Monitoring System using IoT by Dr. R Kamalraj and DR. M Sakthivel.