**My approach:**

As I have to train my model only using the normal Dataset I will use an AutoEncoder based model.

Because in AutoEncoder based models we take a large image and encode it to a very small vector. Using this small vector we reconstruct the original image by upsampling the vector. But in this process we get some errors because our model can't generate the exact same original image which is called Reconstruction Error (RE). So, if we train our AutoEncoder based model using the normal activity of CCTV footage it will create some RE in the normal scenario. If any anomalous event happens the Reconstruction Error (RE) will be higher and our model will detect it as anomaly. In Fig-1 the left side represents the original image and the right side represents the constructed image. As we can see, the model can't reconstruct the image perfectly hence there are some RE.

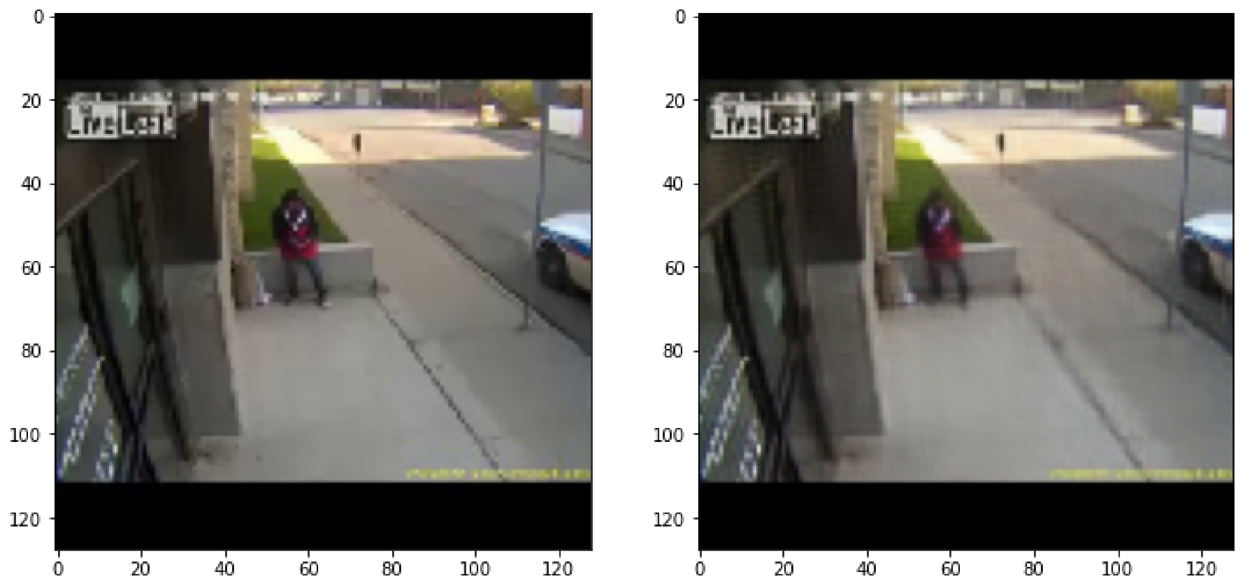

Fig 1: Reconstruction of image using AutoEncoder

We have to set a threshold RE by analyzing the training RE. IF any frame exceeds this RE it will be detected as anomalous activity. The RE is generated by calculating the mean square error(MSE).

Only relying on RE may not be enough for a real time scenario. Following the Article stated in this link (https://arxiv.org/pdf/1901.06355v1.pdf) I applied Kernel Density Estimation (KDE) to check if an image belongs to the normal class or not. In the case of

KDE a latent space is generated using the training images. For anomaly images the distance from the latent space will be higher, on the contrary the distance will be much lower for training images.

The bottleneck layer/the last layer of encoder consists of the highest pixel values of an image. I took this last layer which is the most compressed feature map in order to lower dimensionality of the latent space.

We will classify a frame as anomalous by analyzing both KDE and RE.

**Steps:**

1. At First I took a video where both normal and anomalous events are present. For training purposes I only extracted the normal part from the video. After that 1848 frames were extracted from that video using openCV.
2. I used raw images for training without using any kind of DATA augmentation because my approach is to overfit the model. The more overfitted it will be, the less RE will be generated. I ran through 120 epochs where the training loss < Validation loss in the last epochs.

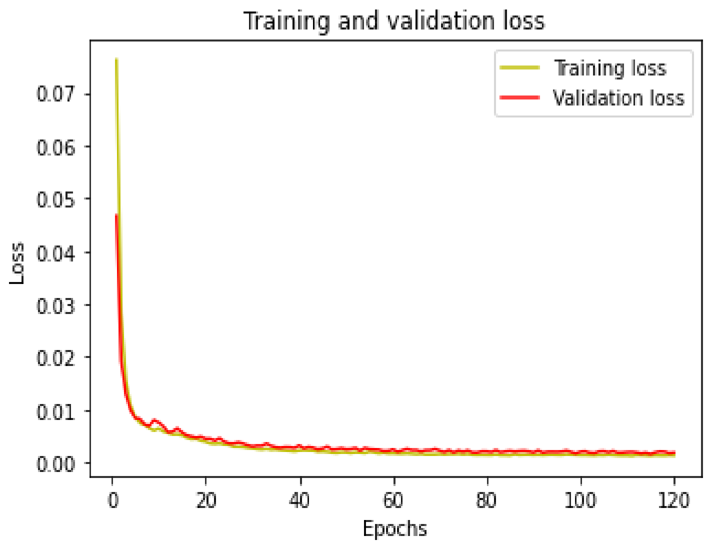

Fig-2: MSE Loss vs Epochs

3. For extracting the features from the input image I made a Custom CNN model in the encoder part and in the decoder part the mirror image of this encoder is used.
4. For setting the RE and KDE threshold I set a relatively higher value of RE and density_threshold from my testing_image RE.

5. In case of any unexpected event the RE value becomes larger than the RE Threshold. If reconstruction_error > reconstruction_error_threshold and density < density_threshold the frame will be anomaly.

**Model training:**

1. I trained several custom CNN models to find the best model in this scenario. As my scenario / training image is complex, I extracted a comparatively big number of features (128) in the first layer of convolution.
2. In the middle bottleneck layer I extracted 8 features in the convolutional layer.
3. As I need to minimize the RE I tried to overfit the model. So I set different epoch batches and layers for creating different models and finally chose the model which is working better in this scenario.
4. If we have different CCTV footage this model can be tuned and we have to find out which model works best for a particular scenario.

**Limitations:**

1. The environment has to be constant. If any new object or new types of incident happened in that constant environment then it will be labeled as anomaly. As surveillance cameras cover only a particular area this approach will work.