

CS 783 VISUAL RECOGNITION

Assignment 4

Unsupervised moving object detection in surveillance videos

Kritik Soman-18104052

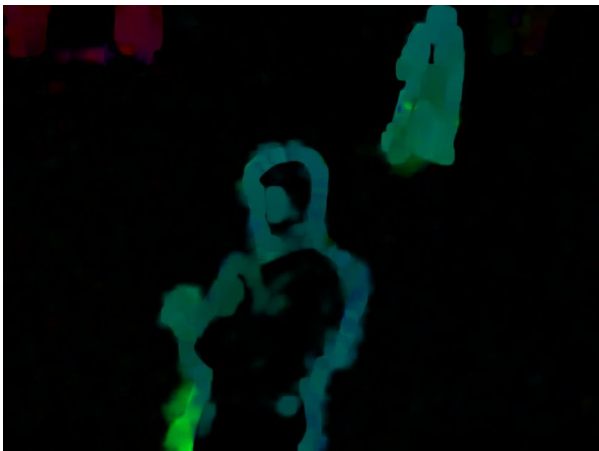
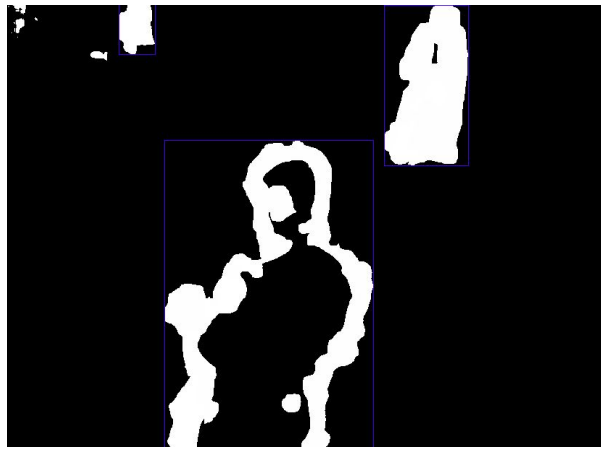
Darshan Ramesh Shet - 16104024

Problem Description:

In surveillance camera recordings, a common vision task is the detection of moving objects. Using a supervised framework can help in accurately detecting a predefined set of objects but generating annotated surveillance videos is a tedious task. Therefore, in this assignment, we made an attempt to solve object detection in videos in a completely unsupervised manner. We combined the traditional computer vision technique (Optical Flow) and deep learning (Autoencoder + Classifier) to provide the bounding box and corresponding class label (example: class 0, class 1 etc.) of all moving objects in the surveillance video. Our algorithm will not name the classes. We used optical flow for detecting motion and blob extraction for localization, followed by an autoencoder + classifier.

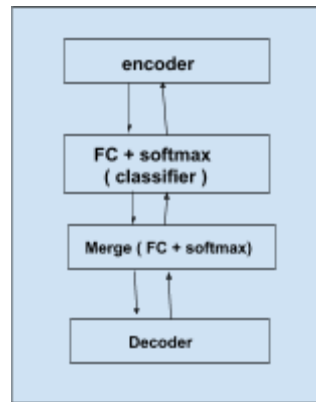
Generating training images

We ran optical flow on three (camera 1, 3 and 5) of the given surveillance videos and created a mask of blobs of the moving objects. Then the contours of the mask were found and blobs whose contour which had no parent and area greater than a threshold were selected, cropped out and saved. We split this cropped image dataset into training and testing with split ratio 0.1.

	
Optical Flow output (Hue represents direction and Brightness represents magnitude)	Mask created from optical flow

Approach 1

The generated cropped images are fed as input to train the autoencoder. An autoencoder is a neural network which attempts to replicate its input at its output. Encoder output is dimensionally reduced features which is flattened and fed to classifier which has set of fully connected layer and softmax layer. The output of last FC and softmax layer is merged and given to decoder to replicate the input. This approach fails to generate a good compressed representation of input at its output which in turn degrades the classification accuracy.



Network model summary is shown below:

Layer (type)	Output shape	Param #	Connected to
input_14 (InputLayer)	(None, 64, 64, 3)	0	
conv2d_98 (Conv2D)	(None, 64, 64, 16)	448	input_14[0][0]
max_pooling2d_49 (MaxPooling2D)	(None, 32, 32, 16)	0	conv2d_98[0][0]
conv2d_99 (Conv2D)	(None, 32, 32, 16)	2320	max_pooling2d_49[0][0]
max_pooling2d_50 (MaxPooling2D)	(None, 16, 16, 16)	0	conv2d_99[0][0]
conv2d_100 (Conv2D)	(None, 16, 16, 32)	4640	max_pooling2d_50[0][0]
max_pooling2d_51 (MaxPooling2D)	(None, 8, 8, 32)	0	conv2d_100[0][0]
conv2d_101 (Conv2D)	(None, 8, 8, 32)	9248	max_pooling2d_51[0][0]
max_pooling2d_52 (MaxPooling2D)	(None, 4, 4, 32)	0	conv2d_101[0][0]
flatten_14 (Flatten)	(None, 512)	0	max_pooling2d_52[0][0]
dense_50 (Dense)	(None, 128)	65664	flatten_14[0][0]
activation_38 (Activation)	(None, 128)	0	dense_50[0][0]
dense_51 (Dense)	(None, 3)	387	activation_38[0][0]
lambda_13 (Lambda)	(None, 3)	0	dense_51[0][0]
concatenate_13 (Concatenate)	(None, 131)	0	lambda_13[0][0] activation_38[0][0]

dense_52 (Dense)	(None, 128)	16896	concatenate_13[0][0]
activation_39 (Activation)	(None, 128)	0	dense_52[0][0]
dense_53 (Dense)	(None, 512)	66048	activation_39[0][0]
activation_40 (Activation)	(None, 512)	0	dense_53[0][0]
reshape_13 (Reshape)	(None, 4, 4, 32)	0	activation_40[0][0]
up_sampling2d_41 (UpSampling2D)	(None, 8, 8, 32)	0	reshape_13[0][0]
conv2d_102 (Conv2D)	(None, 8, 8, 32)	9248	up_sampling2d_41[0][0]
up_sampling2d_42 (UpSampling2D)	(None, 16, 16, 32)	0	conv2d_102[0][0]
conv2d_103 (Conv2D)	(None, 16, 16, 32)	9248	up_sampling2d_42[0][0]
up_sampling2d_43 (UpSampling2D)	(None, 32, 32, 32)	0	conv2d_103[0][0]
conv2d_104 (Conv2D)	(None, 32, 32, 16)	4624	up_sampling2d_43[0][0]
up_sampling2d_44 (UpSampling2D)	(None, 64, 64, 16)	0	conv2d_104[0][0]
conv2d_105 (Conv2D)	(None, 64, 64, 3)	435	up_sampling2d_44[0][0]
=====			
Total params: 189,206			
Trainable params: 189,206			
Non-trainable params: 0			

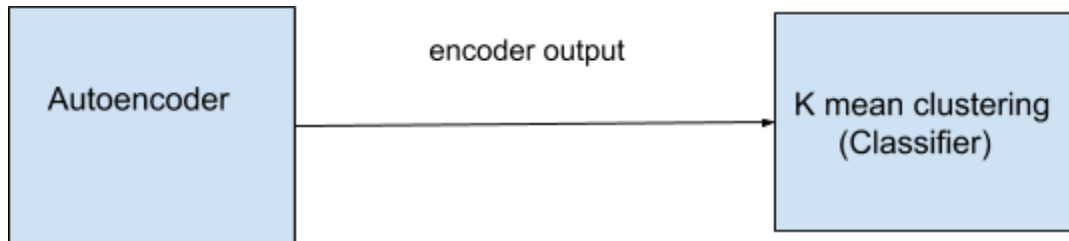
Decoder output (reconstructed input image) :

1st row : Input Image 2nd row : decoder output



Approach 2

Next we tried to train the autoencoder without the classifier(FC + softmax). This improved the quality of the reconstructed image. We used k means clustering on encoder feature for classification. Despite improvement in reconstruction, the feature generated by autoencoder fails to discriminate between classes. So there was not much change in output from the previously tried approach.



Autoencoder model Summary :

Layer (type)	Output Shape	Param #
input_18 (InputLayer)	(None, 64, 64, 3)	0
conv2d_131 (Conv2D)	(None, 64, 64, 32)	896
max_pooling2d_65 (MaxPooling)	(None, 32, 32, 32)	0
conv2d_132 (Conv2D)	(None, 32, 32, 32)	9248
max_pooling2d_66 (MaxPooling)	(None, 16, 16, 32)	0
conv2d_133 (Conv2D)	(None, 16, 16, 64)	18496
max_pooling2d_67 (MaxPooling)	(None, 8, 8, 64)	0
conv2d_134 (Conv2D)	(None, 8, 8, 64)	36928
up_sampling2d_57 (UpSampling)	(None, 16, 16, 64)	0
conv2d_135 (Conv2D)	(None, 16, 16, 32)	18464
up_sampling2d_58 (UpSampling)	(None, 32, 32, 32)	0
conv2d_136 (Conv2D)	(None, 32, 32, 32)	9248
up_sampling2d_59 (UpSampling)	(None, 64, 64, 32)	0
conv2d_137 (Conv2D)	(None, 64, 64, 3)	867
Total params: 94,147		
Trainable params: 94,147		
Non-trainable params: 0		

Decoder output (reconstructed input image) :

1st row : Input Image 2nd row : decoder output



Final Approach

Finally we decided to use the feature from VGG pretrained network to fit a k means model with number of clusters = 3. There was significant improvement in result from previous approaches. We observed that cropped images of people, car and background were clustered into a separate classes.

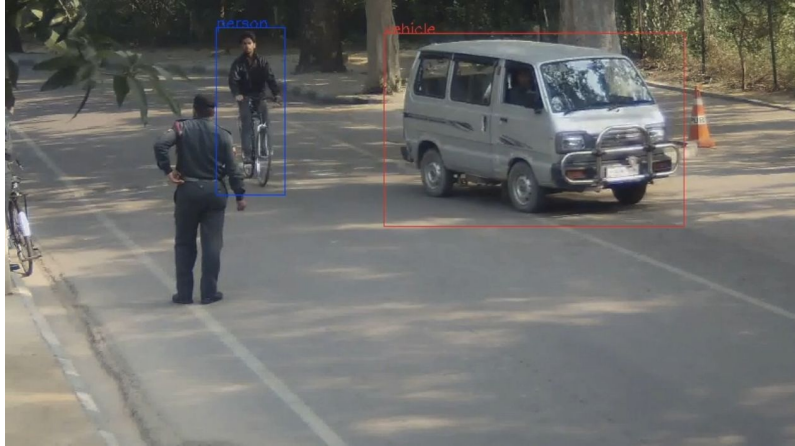
MD5 hash : 3cd7f738ebc2419a764fa3e556bb6e78 joblib_model.pkl

Testing

We ran optical flow on the test video (camera 2) and got the bounding boxes of all the blobs with area greater than a threshold and which has no parent contour. We extracted the VGG feature of the bounding box and showed the bounding box with corresponding cluster. Since most images of cluster 0 belonged to people and cluster 1 belonged to car, for visualization we displayed the label 'person' and 'vehicle' in the test video. Since cluster 3

Result screenshot

Class 0 is shown as person and class 1 is shown as vehicle for visualization.



Drawbacks of our approach

- Optical flow only detects changes in intensity level of image for detecting motion. If there is change in lighting, false bounding boxes would be shown.
- Movement of shadow is also at times detected as an object.
- People on cycles, motorcycles and pedestrians are clustered into same group.
- Single object is at times put into multiple bounding boxes.
- If an object stops moving, it is no longer tracked.