ECE-6360-COMPUTER VISION

To: Dr. Sunanda Mitra

From: Tsung-Sheng Huang, Sharanya Ramakrishna, Supriya Ramaswamy

Date: 20th April 2018

Title: Assignment #4

ABSTRACT:

The main objective of this assignment is to implement 3D SIFT using the 'Bag of words' approach for classification of different actions. In this case we have chosen three actions i.e walking, running and jumping. We start by randomly selecting interest points, then we compute the descriptors, we find clusters within different frames in the videos and the frequency of their co-occurrence to create the grouping histograms which is then used to train a classifier (we have used SVM) for classification of actions.

INTRODUCTION:

- The actions obtained from video data inherently contain spatio-temporal information, meaning we need descriptors which can encode this kind of information for classification of action videos.
- Using 'bag of words' approach in 3D SIFT implementation, the third-dimension time is also used for classification of actions.
- Previous methods that have used 'bag of words' approach have used only simple features like gradient magnitude for feature detection. But this feature alone cannot completely and explicitly describe spatio-temporal nature of the data [1].
- Hence in the paper studied and implemented by us, by *Scovanner, Paul, Saad Ali, and Mubarak Shah. "A 3-dimensional sift descriptor and its application to action recognition." Proceedings of the 15th ACM international conference on Multimedia. ACM, 2007,* they have used the magnitude(mx,y) and 2D-orientation(θ) along with Φ in 3D space , where Φ is the angle away from the 2D orientation direction.
- The key-points are selected at random and descriptors are then created by sampling sub-regions around these keypoints.
- For each of these sub regions we accumulate the orientations into a histogram.
- These descriptors gathered are then quantized by clustering them into some pre-specified number of clusters (in this case = 50) using pairwise distance between keypoints' features and using linkage between them. K-means algorithm is then used for finding clusters by using the center points (words) from the previous step.
- In each of the 50 clusters we find the contribution of keypoints from each video in each cluster. This gives signature of each video in the 50 clusters.

- After we obtain the signature of keypoints, we create a co-occurrence matrix that gives how many times a particular word has occurred with other words, and this can be used to find more discriminative representation of features.

- To quantify the similarity between any two words we create the co-relation matrix. If the correlation is above a specified value (in our case 0.95) we add those frequencies and combine them to create a grouping histogram.

- We then train a classifier (SVM in this case) to classify each of our actions and test its accuracy.

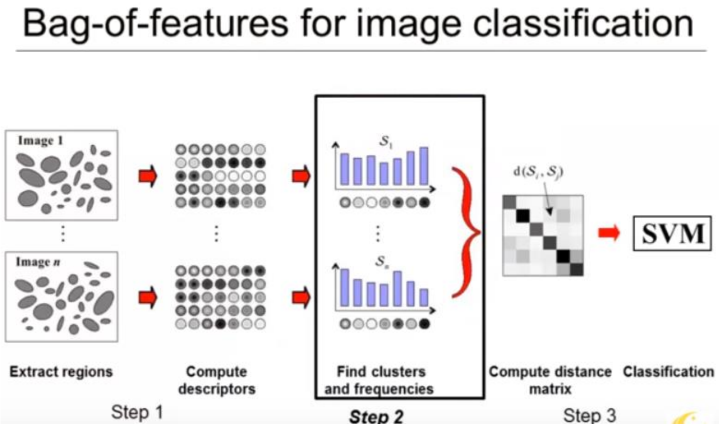

Figure 1 : Steps for classification using bag of words

3D SIFT descriptors :

- In a 2D pixel we define the gradient magnitude and orientation as follows ,

$$m_{2D}(x,y) = \sqrt{L_x^2 + L_y^2}, \quad \theta(x,y) = \tan^{-1}(\frac{L_y}{L_x}).$$

Where, $L_x = L(x+1,y,t) - L(x-1,y,t)$

$L_y = L(x,y+1,t) - L(x,y-1,t)$

- In 3D pixels we define,

$$m_{3D}(x,y,t) = \sqrt{L_x^2 + L_y^2 + L_t^2},$$
$$\theta(x,y,t) = \tan^{-1}(L_y/L_x),$$
$$\phi(x,y,t) = \tan^{-1}(\frac{L_t}{\sqrt{L_x^2 + L_y^2}}).$$

Where, $L_t = L(x,y,t+1) - L(x,y,t-1).$

- Since pixels in a video contain data of spatio-temporal nature, we need to encode it using 3 dimensions, hence we use x,y and t as the three dimensions.

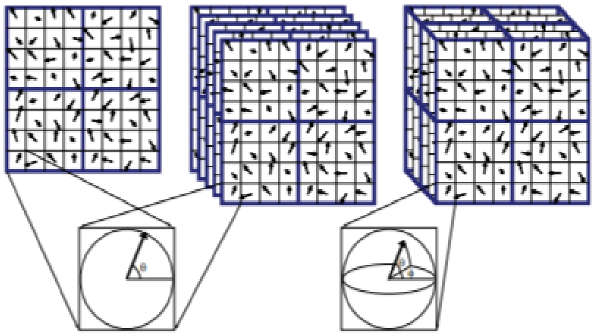

Figure 2 : Right image shows 3D sub volumes

- The next step is to obtain the 3D SIFT descriptor.

- We start by calculating orientation sub histograms. For this we first need to rotate the 3D neighborhood around the keypoint[1]. This is done by taking each (x,y,z) for each neighborhood pixel and multiplying it with the given matrix[1]

$$\begin{bmatrix} \cos\theta\cos\phi & -\sin\theta & -\cos\theta\sin\phi \\ \sin\theta\cos\phi & \cos\theta & -\sin\theta\sin\phi \\ \sin\phi & 0 & \cos\phi \end{bmatrix}$$

Figure 3 : Orientation matrix

- The sub-regions are obtained by sampling the neighborhood area around the interest point [1]

- For each 3D sub region, we need to accumulate the orientations in a histogram. The final descriptor is a vectorization of these sub histograms [1]

CLASSIFICATION USING SUPPORT VECTOR MACHINE:

The classifier must be trained using a set of negative and positive training image examples. The classifier “learns” the regularities in the data. If training was successful classifier can classify an unknown example with a high degree of accuracy.

Task of separating classes in feature space is called binary classifier.

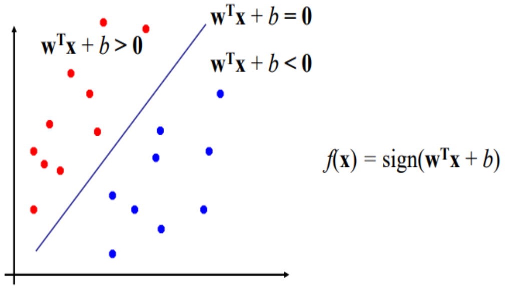
$\mathbf{w}^T\mathbf{x} + b > 0$
$\mathbf{w}^T\mathbf{x} + b = 0$
$\mathbf{w}^T\mathbf{x} + b < 0$
$f(\mathbf{x}) = \text{sign}(\mathbf{w}^T\mathbf{x} + b)$

Figure 4 : Binary SVM classifier

Linear Classifiers: We can draw many linear separators as shown below but we need to find the optimal linear separator so that there are no errors while distinguishing between the two classes.

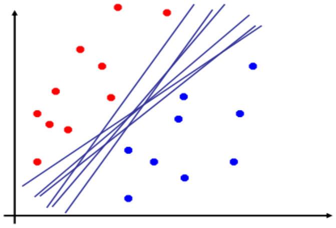

Figure 5 : Finding optimal linear separator

Point to Plane Distance Equation

Once the optimal linear separator is found we find the distance from the nearest point to the separator. Distance from point to the separator is found using the following equation

$$r = \frac{\mathbf{w}^T\mathbf{x} + b}{\|\mathbf{w}\|}$$

Points closest to the hyperplane are called as the support vectors. Margin 2γ of the separator is the width of separation between the two classes.

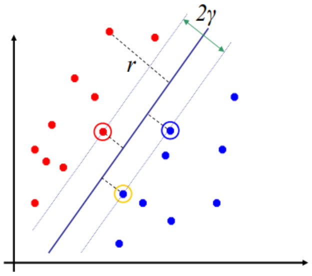

Figure 6 : Finding margin using hyperplaneIt’s good to maximize the margin intuitively and this implies that only support vectors are important, other training examples can be ignored.

ALGORITHM:

1. **Reading Video and 3D SIFT:**

Select any three actions to be used and store frames from each video in a 3D matrix of rows, columns and time and select 200 keypoints from each video using Create_Descriptor . Each keypoint will generate a feature vector named 'ivec' in our case which will have 1X640 dimension for each keypoint. Since we have 200 keypoints and total 29 videos (total of all 3 actions) we have 29X200X640 = 5800X640 dimension for the ivec.

2. **Hierarchical K-means**:

Using pairwise distance between these keypoints in the 3 videos, we obtain the required 50 clusters as specified.

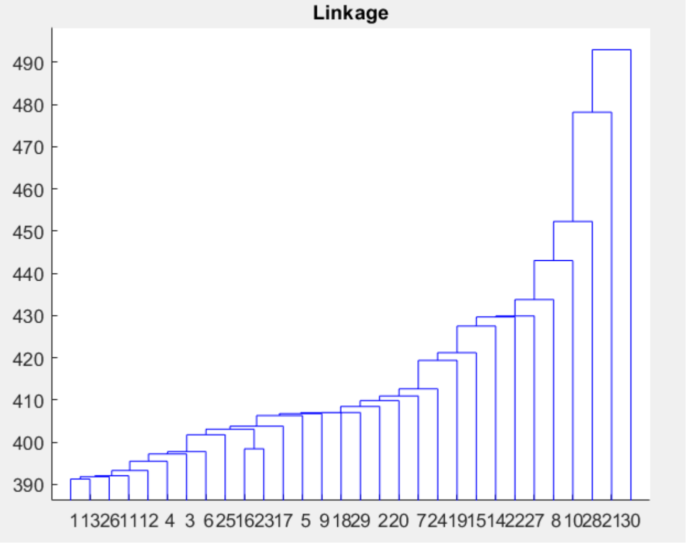

Figure 7 : Linkage obtained using pairwise distance between each of the keypoints

Using K-means to find the cluster for each keypoint by using the initial center which is obtained by hierarchical K-means. Each of these clusters are called 'words' (words in terms of feature vectors which are descriptors around each keypoint).

3. **Signature for each video:**

Since we know first 200 X 10 keypoints belong to action 1, 2nd 200 X 9 keypoints belong to action2 and final 200X10 keypoints belong to action 3 and we have the cluster number associated with the 5800 keypoints we can create a matrix H that gives signature of each video i.e. how many keypoints from each of the videos belongs to each cluster.

4. **Feature grouping Histogram:**

   If we find two similar words, it means these two words are capturing something similar and hence can be used for distinguishing between actions. For quantifying this we calculated the correlation matrix between the words and if the correlation value was above 0.95 we added the two cluster frequencies to obtain a grouping histogram.

5. **SVM Training and 'Leave one out':**

   In SVM, we are not able to train the multi-class problem, so we separate the class into "Walk" and "Not Walk". In our case, we have 29 training samples, every time we train with 28 samples and test on only one data. Then we keep doing this process for 29 times (Take out different sample each time for testing). We tested two different kinds of training samples, the first one is the signatures of the videos without grouping which is the output of "histcounts". The other one is the grouping histogram which is generated from feature grouping histogram. The error rates of these two are 0.1724 and 0.1379 respectively. The error of the model trained using grouping histogram gives us a better error rate.

CONCLUSION:

- Hence, we studied and implemented 3D SIFT for actions classification using *Scovanner, Paul, Saad Ali, and Mubarak Shah. "A 3-dimensional sift descriptor and its application to action recognition."*

- Feature detection for classification was done using 'bag of words' approach to obtain spatio-temporal encoding at pixel level.

- Hierarchical K-means approach was used for finding the centers of 50 clusters and then apply the K-means classify the keypoints into 50 clusters using the initial centers.

- Correlation between words was obtained to generate the correlation matrix and this correlation matrix was used for creating the grouping histogram.

- The grouping histogram has a better performance than using the signature (histcounts) without grouping.

REFERNCES:

1. Scovanner, Paul, Saad Ali, and Mubarak Shah. "A 3-dimensional sift descriptor and its application to action recognition." Proceedings of the 15th ACM international conference on Multimedia. ACM, 2007.

2. https://www.youtube.com/watch?v=iGZpJZhqEME&index=17&list=PLmyoWnoyCKo8epWKGHAm4m_SyzoYhslk5. Fundamentals of computer vision: Mubarak Shah