Action Recognition with Optical Flow

Ammar Raşid S017288

Obtainign optical flow vectors

Instead of using a key-point (corner) detector and use only key points for optical flow, we used a grid of a customizable step size between subsequent pixels in the grid. We used a step size of 5. We ran Lucas Kanade method for optical flow estimation between every two subsequent frames in the sample video. Lucas-Kanade gives us the new positions of selected points in the new frame. By subtracting the positions, we obtain the optical flow vector of this point $\nu = [x, y]^T$, with direction $\theta = tan^{-1}(\frac{y}{x})$.

Figure 1: Caption

HOOF: Histogram of Optical Flow

We used different numbers of bins to explore the performance of different configurations of HOOF feature vector. We tried the values from 2 to 5. We first made sure that $0 \geq \theta \leq 2\pi$. That was done by scaling values larger than 2π, and adding $2n\pi$ to negative values of theta, where $n = 2\pi(\lfloor \frac{\theta}{2\pi} \rfloor + 1)$. One sample per class was drawn out for testing.

PCA: Principal Component Analysis

First, we obtained the covariance matrix of HOOF matrix $N \times B$, where N is the number of training samples. We then computed the eigenvectors, and their corresponding eigenvalues of the covariance matrix. We keep adding eigenvectors one by one to our projection space, that is, concatenating them in our projection matrix, until the summation of their corresponding eigenvalues reach the required energy (variance) level. This way we can represent the data with less dimensions, without losing much information, precisely at most 10%.

KNN: K-Nearest Neighbour

After we have obtained a feature space that represents the action in each scene using HOOF, we calculate the distance between the inspected test instance and all other training instances, and return the labels of the nearest K of them, that is, the ones with least euclidean distance.

$$Euclidean(a, b) = ||a - b|| = \sqrt{\sum_{i}^{n} (a_i - b_i)^2} \tag{1}$$

Where n, is the number of dimensions.

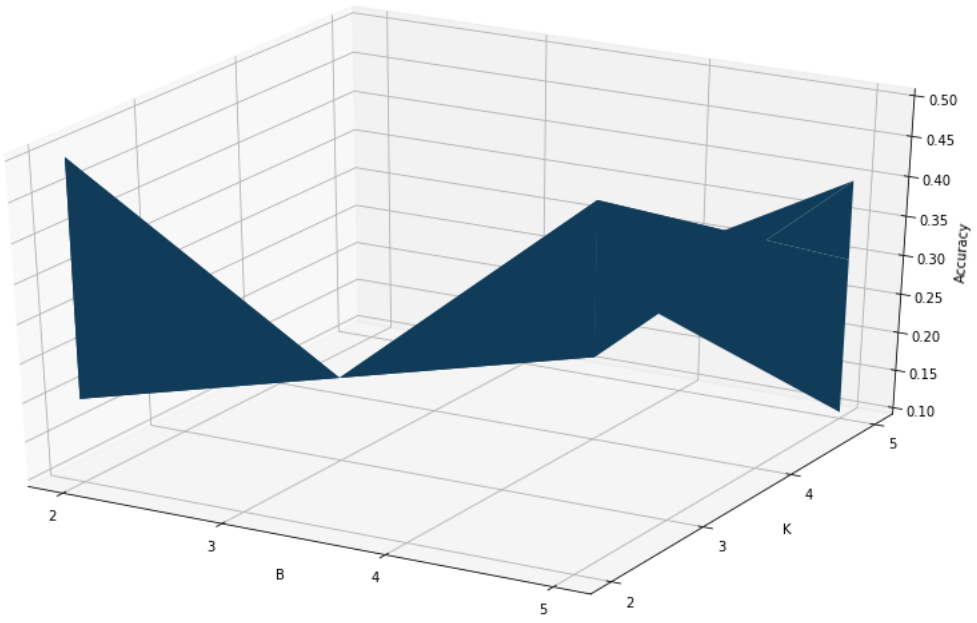

Figure 2: Mesh of B vs K vs Accuracy

Grid search

We ran the pipeline on a grid of parameters, namely, K in KNN, and B (number of bins) in HOOF. The values we used were $\{2, 3, 4, 5\}$ for each of the parameters. Runtimes ranged between 140 and 180 seconds on a double 1.4 Ghz actual cores processor, and 4 virtual ones. Best model was with $B = 2$ and $K = 3$, with 0.5 accuracy. Average accuracy over all parameters combinations was 0.2815, with standard deviation of 0.107.

1 Conclusion

We can conclude that too large K doesn't guarantee better classification performance as the KNN model will start considering further neighbours. Small number of bins makes the optical flow directions much less sophisticated, yet that still resulted in a better feature representation that achieved the best accuracy.