CMPUT 428 Final Project Proposal: Learning Based Real-Time Object Tracking through Motion Vector Predictions on Motion Blur

Jiachen Huang Mingze Sun
jiache19@ualberta.ca mingze9@ualberta.ca

Introduction

In the rapidly advancing field of computer vision, object tracking stands out as a pivotal challenge, particularly in dynamic environments where precision and reliability are paramount. The Sum of Squared Difference (SSD) tracker has established itself as a robust framework for detecting objects in images especially when integrated with the Lukas-Kanade algorithm. However, its application in real-time scenarios have significant limitations, notably in handling motion blur. Motion blur is a frequently occurring effect in video recording, arising whenever an object is moving relative to the camera during the exposure time. Its strength depends on the duration of the exposure as well as on the speed of the object relative to the camera.

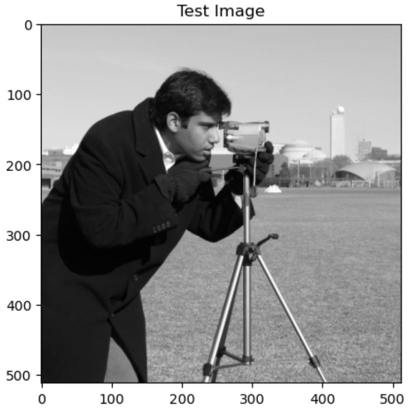
This project proposes an innovative solution aimed at bridging the gap between the need for accuracy and the complexities introduced by motion blur. We intend to develop a methodology that harnesses the power of Convolutional Neural Networks (CNN) to predict motion vectors based on the degree of blurriness observed in the last frame of a video sequence. By training a model that can intelligently infer the trajectory of an object's motion solely from the blur it exhibits, we can significantly enhance the prediction accuracy for subsequent frames.

The cornerstone of this approach is a two-pronged process. Initially, a dedicated CNN model will be trained on a diverse dataset of blurred images, learning to correlate the extent of blurriness with the underlying motion vectors that caused it. Subsequently, this trained model will be deployed as a predictive tool in a real-time tracking system, enabling it to anticipate the position of an object with heightened precision, thereby mitigating the drawbacks of current SSD-based trackers.

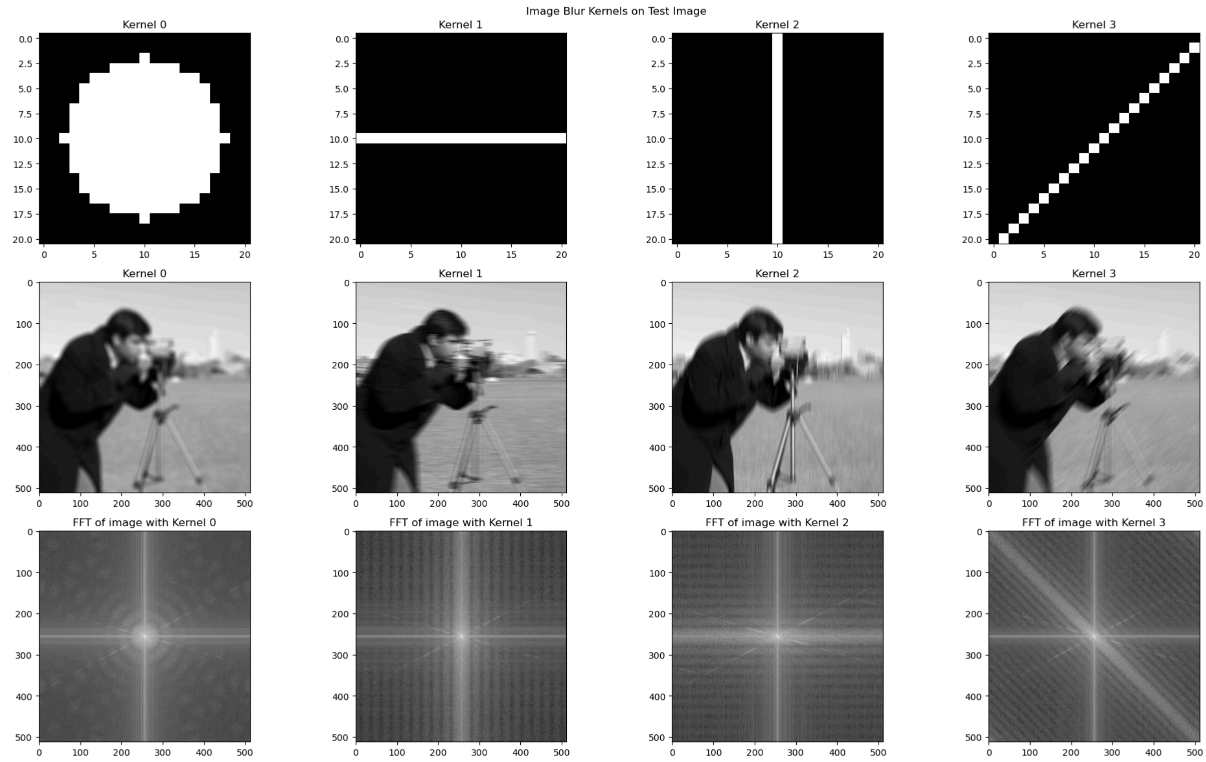
Methodology

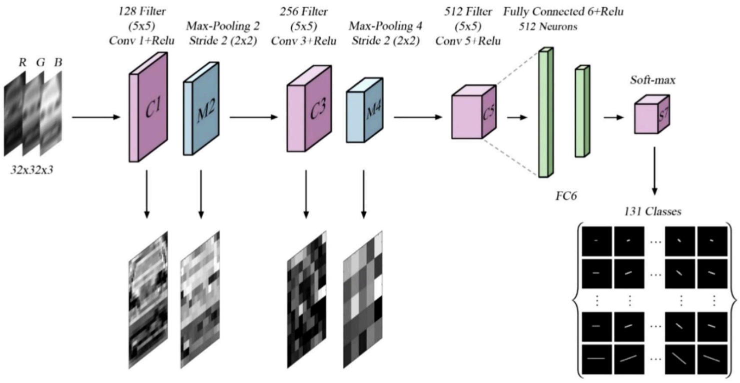
We plan to use the opensource CIFAR-100 dataset and apply motion blur kernels to each image.

original image

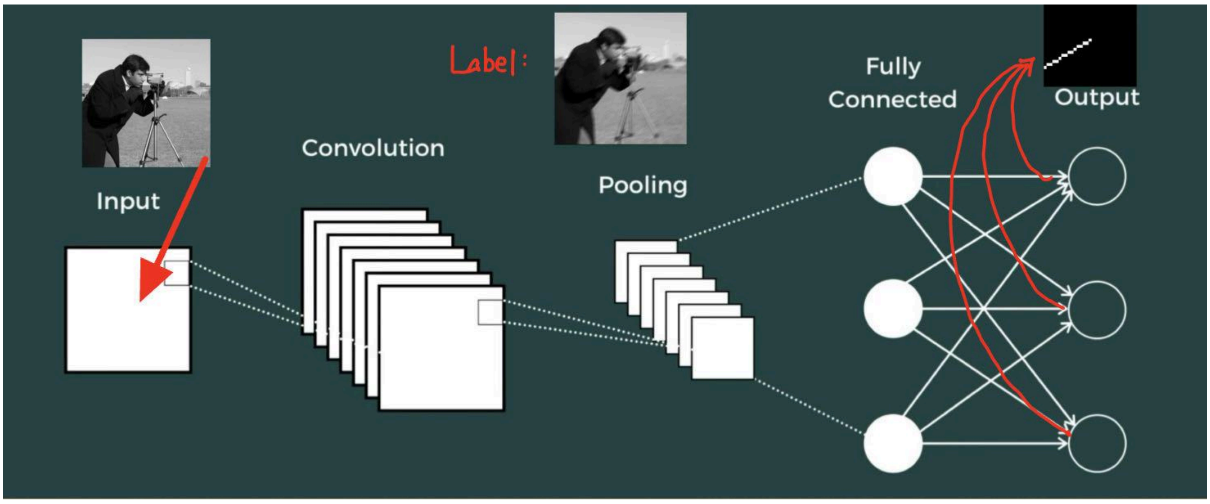


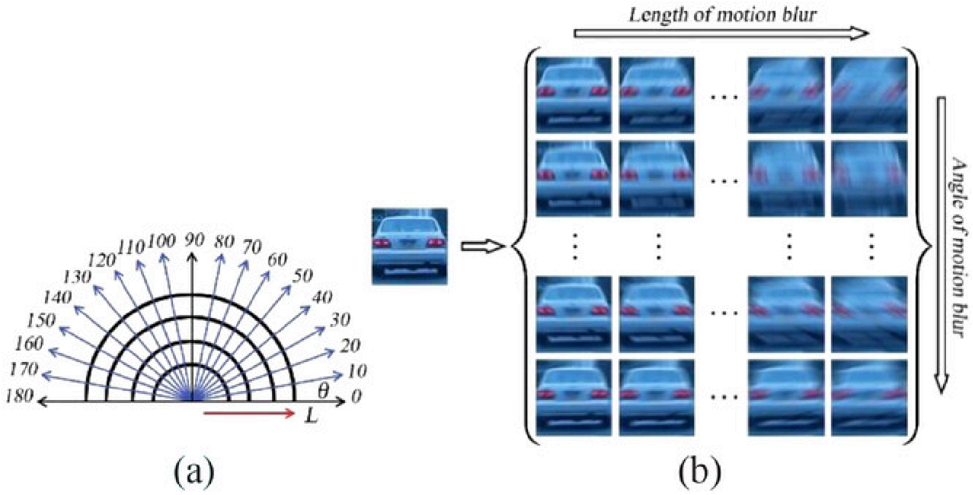
applied blur kernel





In other words, the proposed method presents a model prediction and estimates the location of the blurred object in the next frame according to the motion model of the current frame.





We will be generating 270 different motion kernels based on the motion direction θ and magnitude l.
The range of motion length was discretized into 15 samples from l = 1 to 30 with the interval of 2, and the range of motion angle was discretized into 19 samples from θ = 0° to 180°, and since degree 0 and 180 are equivalent, n_θ = 18.
Therefore, we have n_l * n_θ = 15*18 = 270 number of motion kernels.

The probabilities of 270 motion kernel candidates are predicted in the output of our CNN model, but during the tracking, there may exist more different angles that object may blur with them, so we need to extend the number of motion blur kernels estimated by CNN. Thus, the probabilities of motion kernels for the rotated patches (Left Hand Side) of every image can be estimated by feeding the rotated patch into CNN in Figure 4.

$$\boldsymbol{P}(\mathbf{m} = (l, \theta - \rho) \,|\Psi(\mathrm{I})) = \boldsymbol{P}\left(\mathbf{m} = (l, \theta) \,|\Psi\left(R_\rho \mathrm{I}\right)\right)$$

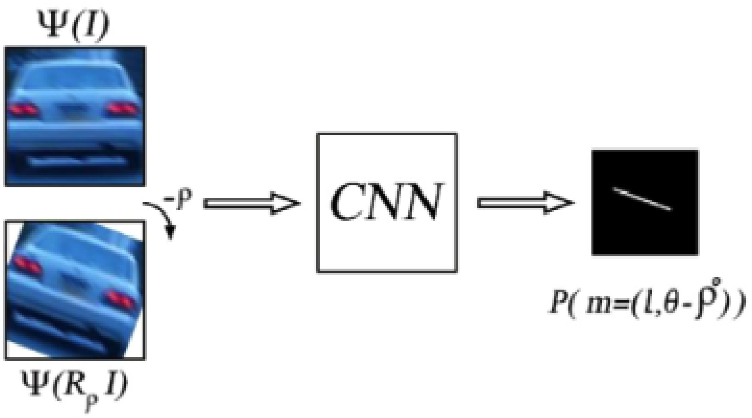


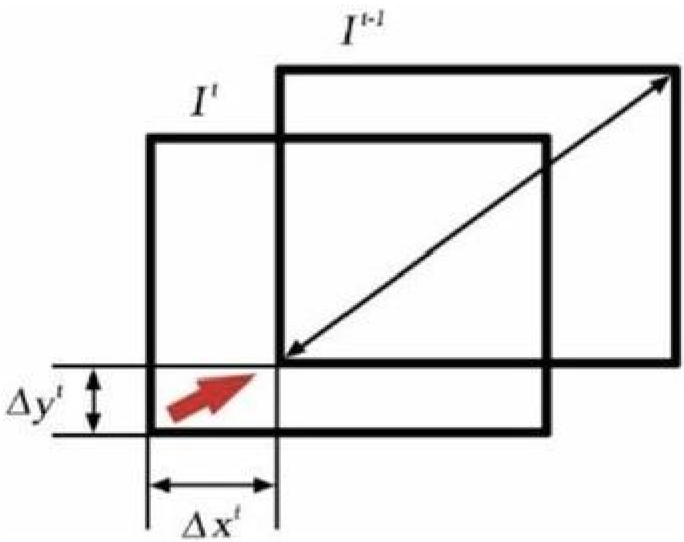
FIGURE 4 Expanding of motion kernel estimation by rotated images. I is a blurred patch and R_ρI is the rotated image

If we assume that there is the same motion distribution on the pixels of every image patch, the confidence of the motion kernel among multiple estimated motion probabilities needs to be defined by:

$$C\left(m_p = (l,\theta)\right) = \frac{1}{Z} \sum_{q:\, p\in\Psi_p} G_\sigma\left(\left\|x_p - x_q\right\|^2\right) P\left(\mathbf{m} = (l,\theta)\,|\Psi_q\right) \tag{5}$$

$$\min_M \sum_{p\in\Omega} \left[-C\left(m_p = (l_p,\theta_p)\right)\right] + \sum_{q\in N(p)} \lambda\left[\left(u_p - u_q\right)^2 + \left(v_p - v_q\right)^2\right] \tag{7}$$

According to Equation (7), the motion vectors of m_p and m_q in Cartesian coordinates are defined as $(u_p - v_p)$ and $(u_q - v_q)$,



Citations

1. “A motion parameters estimating method based on deep learning for visual blurred object tracking” https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/ipr2.12189
2. “How to apply Motion Blur to Images ” https://medium.com/@itberrios6/how-to-apply-motion-blur-to-images-75b745e3ef17
3. “Lucas-Kanade 20 Years On: A Unifying Framework” https://www.ncorr.com/download/publications/bakerunify.pdf
4. “Multiple View Geometry In Computer Vision” https://www.robots.ox.ac.uk/~vgg/hzbook/

5. “Learning a Convolutional Neural Network for Non-uniform Motion Blur Removal”
   https://arxiv.org/pdf/1503.00593.pdf