OpenCV IDEA: Lightweight Optical Flow Model

Aser Atawya

April 4, 2023

Summary of the Proposal

Optical flow is the problem of estimating the motion of objects in an image or video sequence. Optical flow is pivotal to many computer vision applications such as object tracking, video stabilization, and motion analysis, providing essential information about the dynamics of a scene or an object. With the upsurge in the use of deep neural networks, DNN models were developed to solve the optical flow problem, achieving state-of-the-art results by learning to estimate motion from image data. However, DNNs can become computationally expensive making it infeasible to be deployed on embedded systems. This problem is tackled by the development of lightweight DNN models. However, OpenCV model zoo hasn't had any implementation of an optical flow lightweight model yet. Thus, this project aims to find the best lightweight optical flow model in terms of model size, speed, and accuracy to introduce to the OpenCV model zoo.

Background Research in the Optical Flow Problem

The optical flow problem is a fundamental computer vision challenge that involves estimating the motion of objects in a sequence of frames over time. Specifically, it requires calculating the displacement vectors of points in an image from one frame to the next, which can be used to track the movement of objects in the scene. Accurately solving the optical flow problem is challenging due to complex motion patterns and the need for robust methods to handle noise and occlusions. For instance, there is the aperture problem, which refers to the ambiguity that arises when estimating the motion of objects in a scene using only a limited aperture of visual information. Specifically, when a moving object is observed through a small aperture, such as a narrow window or a small sensor, the motion of the object perpendicular to the aperture cannot be estimated accurately. This is because the information available through the aperture is limited to the projection of the object's motion along the aperture direction, resulting in an ill-posed problem. Optical flow estimation has important applications in various areas such as object tracking, video compression, and autonomous navigation.

Optical flow can be divided into dense and sparse. In dense optical flow, a flow vector is calculated for every pixel in an image, resulting in a dense field of flow vectors. This provides detailed information about the motion of every pixel in the scene and is useful for applications such as motion-based segmentation and tracking. Sparse optical flow, on the other hand, involves computing the flow vectors only at specific points of interest in the image, such as corner points or feature points. This results in a sparse set of flow vectors and is less computationally expensive than dense optical flow. Sparse optical flow is typically used in applications where detailed information about motion is not required, such as visual odometry and image stabilization.

Overview of the Optical Flow Problem Solutions

Before deep neural networks, the traditional methods for tackling the optical flow problem included differential, correlation-based, variational, and feature-based methods. Differential methods compute image gradients and solve partial differential equations to estimate the flow field, while correlation-based methods use cross-correlation to find the best match between image patches. Variational methods formulate the optical flow problem as an optimization problem to minimize an energy functional, while feature-based methods track specific features and estimate the flow field by computing their displacements between frames.

Deep neural networks have been applied to solve the optical flow problem by estimating optical flow directly from image pairs, without requiring any hand-crafted features or assumptions about the underlying motion patterns. These networks typically take two consecutive frames as input and output a dense optical flow field as an output. Various CNN architectures have been proposed for this task, such as FlowNet, RAFT, and PWC-Net. These deep learning-based methods have shown significant improvements in accuracy compared to traditional methods, particularly for challenging scenarios such as large displacements and occlusions.

Problem Definition

Even though DNN models have proved to be extremely useful in solving the optical flow problem, they face two important challenges; they require a large amount of training data and can be computationally expensive for real-time deployment. To tackle the former, some of the datasets commonly used in research papers to train the datasets are MPI-Sintel, ChairsSDHom, KITTI, and FlyingThings3D. And to tackle the latter, efficient, lightweight optical flow models, such as LiteFlowNet, SpyNet, and STaRflow are used to account for speed considerations and achieve real-time performance on mobile and embedded devices. Regardless, OpenCV model zoo lacks any implementation of neither any of the aforementioned models nor any other lightweight model despite the importance of the optical flow problem in computer vision. The problem that this proposal is tackling is finding the best lightweight optical flow model in terms of model size, speed, and accuracy to add to the OpenCV model zoo and implementing that candidate using OpenCV DNN for inference while making sure the license for the model allows this adaption. Furthermore, the problem entails analyzing a possible model quantization to further reduce the best candidate's model size and increase its model speed while maintaining reasonable accuracy.

Survey of Lightweight Models

The development of lightweight models entails different techniques and principles to address the optical flow problem with less computation time and power and limited number of parameters. LiteFlowNet, for instance, prioritizes feature extraction and feature wrapping over image wrapping, which is used by FlowNet (Hui, et al. 2018). It employs pyramidal feature extractions, and at each pyramid level, LiteFlowNet optimizes flow inference through a lightweight cascaded network. LiteFlowNet2 was a significant upgrade to LiteFlowNet; it outperformed FlowNet while being 25.3 times smaller in the model size and 3.1 times faster in the running speed (Hui, et al. 2020). LiteFlowNet2 has 2.3 million parameters compared to 160 million parameters for FlowNet and 5.37 million parameters for LiteFlownet (Godet, et al. 2020). Also, there is STaRFlow model, which is based on a double recurrence over both space and time in contrast to LiteFlowNet. It invokes a spatiotemporal recurrent cell repeatedly in a coarse-to-fine scheme for both optical flow and occlusion detection. STaRFlow has 4.77 million parameters. (Godet, et al. 2020) Another model worth mentioning is SPyNet. While SPyNet has 1.2 million parameters compared to 39.16 million parameters for FlowNetC, a lightweight version of FlowNet, FlowNetC actually runs 4 times faster than SpyNet (Ranjan and Black, 2016). So, it's important to note that model size and speed are not strictly proportional. Also, model size extends beyond the number of parameters to also include the depth of the model as expressed by the number of layers. To ensure a fair analysis of the different lightweight models, we need to precisely define the metrics that will be used to evaluate the model performance against.

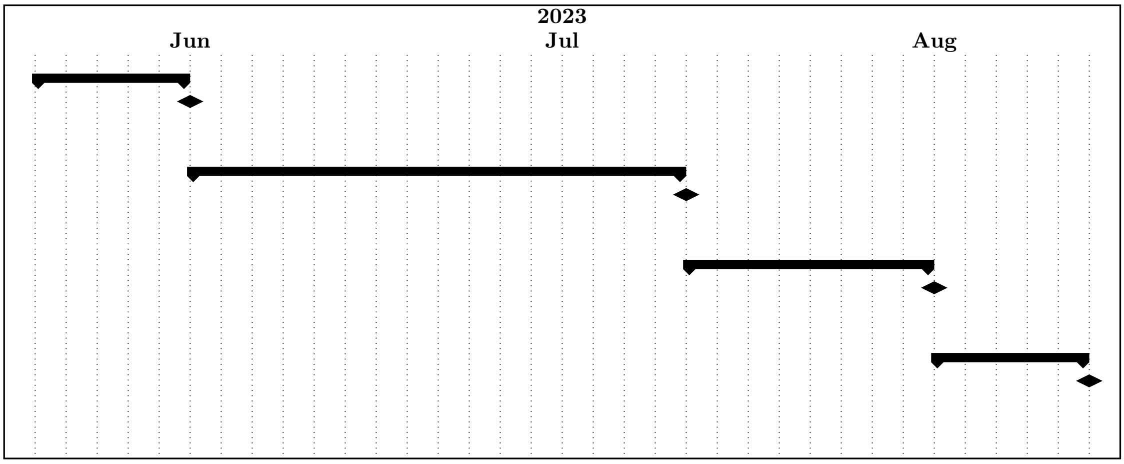
Evaluation of Model Performance

Evaluating deep neural network optical flow models against metrics such as speed, model size, and accuracy is essential to determine their performance and suitability for different applications. Speed can be measured by computing the average time taken to process each frame or image in a video sequence (latency), or by measuring the number of frames per second that the model can process (throughput). Model size can be measured by counting the number of parameters in the model or by measuring the size of the model file. Model size reflects the resources needed to store and deploy the model, with smaller models being more efficient and easier to use in resource-constrained environments and so is more desired by OpenCV. Finally, accuracy measures how well the model can estimate the true motion in a scene, providing a measure of the model's overall performance. Accuracy can be measured using various evaluation metrics such as mean squared error (MSE), endpoint error (EPE), and percentage of pixels with errors (PPE) with endpoint error being the prevalent metric used in evaluating optical flow models.

While other metrics such as robustness to occlusions and illumination changes can also be evaluated by testing the model on challenging datasets with these properties, the three most important metrics are speed, model size, and accuracy. These three metrics can be computed on a validation dataset, where the ground-truth optical flow is available to calculate accuracy, or on a benchmark dataset such as Sintel or KITTI. Fortunately, the MPI Sintel Flow Dataset web page provides the measured accuracy, in terms of endpoint error(EPE), of 421 different optical flow models on the MPI Sintel Flow benchmark dataset.

Furthermore, the endpoint error results presented on the MPI Sintel Flow web page are broken down to multiple metrics depending on velocities, closeness and farness from occlusion boundaries and frames' adjacency. I think it's important to evaluate the model based on multiple types of errors to detect bias instead of using the overall error as the only metric. Evaluating different lightweight models against these metrics allows for informed comparisons and decisions when selecting the best candidate to introduce to the OpenCV model zoo. Also, the same metrics will be used to assess the effect of model quantization and so determine its feasibility. Now that we defined the performance metrics, we can present a timeline for the expected progress to achieve the expected outcomes.

Tentative Timeline



Phase 1: This is the research-focused phase. I will start this phase by surveying lightweight models to shortlist the best three models depending on their performance metrics in terms of speed, model size, and accuracy on the MPI Sintel Flow dataset as well as whether their license allows borrowing them for OpenCV Model Zoo. Afterwards, I will analyze the performance of the three shortlisted models on different datasets to avoid being biased by the MPI Sintel dataset. Depending on the cumulative results of the three models on different datasets, I will pick the best candidate.

Phase 2: This is the longest and the most important phase as well as the heavy-coding phase. I will implement the best candidate from phase 1 using OpenCV DNN for the inference. As part of the implementation, I will check if any necessary patches need to be added to OpenCV to support the new model. To validate my model implementation, I will run the model on the benchmark datasets, such as MPI Sintel dataset, and evaluate the results of my implementation against the model's original results. I will continue debugging my implementation until my implementation results match those of the original model.

Phase 3: To further optimize the best candidate for OpenCV, I will experiment with multiple quantization techniques to see whether the best candidate model can be quantized without a significant drop in accuracy. Some of the possible quantization techniques in mind are post-training dynamic range quantization and mixed-precision quantization. I will evaluate the quantized model on different benchmark datasets using the same performance metrics to determine the feasibility of the quantization technique.

Phase 4: This is the last phase where I wrap the project together. I will optimize the evaluation of the best candidate model by evaluating the model against other metrics such as robustness to occlusions and illumination changes. Also, I will devise examples in C++ and Python to demonstrate the use of the new model.

References

[1] Tak-Wai Hui, Xiaoou Tang, and Chen Change Loy. LiteFlowNet: A lightweight convolutional neural network for optical flow estimation. In IEEE Conference on Computer Vision and Pattern Recognition, June 2018
[2] Tak-Wai Hui, Xiaoou Tang, and Chen Change Loy: A lightweight optical flow cnn—revisiting data fidelity and regularization. In IEEE transactions on pattern analysis and machine intelligence 4, March

2020
[3] Pierre Godet Alexandre Boulch, Aurélien Plyer, and Guy Le Besnerais. STaRFlow: A SpatioTemporal Recurrent Cell for Lightweight Multi-Frame Optical Flow Estimation. In 2020 25th International Conference on Pattern Recognition (ICPR), July 2020.
[4] Anurag Ranjan and Michael J. Black. Optical flow estimation using a spatial pyramid network. In CVPR, Nov 2016