Object Detection and Variational Optical Flow

Aaron Mei
SEAS 2025
Dr. Jianbo Shi GRASP Laboratory, CIS

Abstract

Computer vision is an incredibly broad field that seeks to extract information from images and video. Researchers have presented a plethora of creative and unique approaches that have been pruned and improved over the years. This work explores some foundational algorithms such as the "You only look once" (YOLO) algorithm, as well as a more antiquated variational optical flow method.

Research Goal

In this project, we aim to implement and understand important algorithms in object detection and optical flow.

Background

Object detection:

The aim of object detection is to label certain objects in a scene, usually by using a bounding box to capture the location. At first, numerous heuristics were employed such as the use of descriptors (e.g., Histogram of oriented gradients, scale invariant feature transform) and feeding them into a support vector machine.

Soon, computationally intensive pipelines such as RCNN were developed along with deformable parts models.

In 2015, Redmon et. al., introduced YOLO (You Only Look Once) which provided fast and accurate object detection by merging bounding box proposals

Optical flow:

The problem of optical flow focuses on tracking how certain objects (sparse optical flow) or how each pixel of an image (dense optical flow)

This project focuses on variational methods which define an energy function that is sought to be minimized. Some more modern methods
have also employed neural networks with great success.

Methods

The two algorithms that this project focuses on are YOLO and a variational optical flow method proposed by Brox et. al.

YOLO:

The YOLO algorithm was implemented in PyTorch and trained using cloud computing on Google Collaboratory. To study the YOLO algorithm under a controlled environment, we implement a shape framework to randomly generate blank images with colored ellipses and triangles.

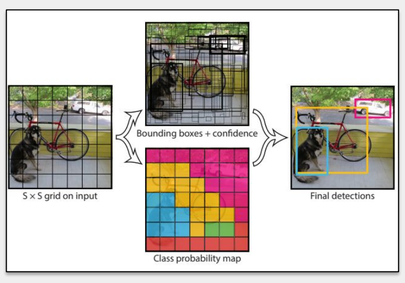
Due to the relative simplicity of the task and to speed up training, we reduce the size of the fully connected layer from 4096 nodes to 496 nodes.

Brox optical flow:

The Brox algorithm for optical flow was implemented in python with libraries including numpy and scipy. The shape framework was used to generate shapes that moved to different positions a couple of pixels away between frames. The algorithm then calculated the flow for each pixel in the image, which was plotted as a vector field with matplotlib. Modifications to the algorithm included replacing the iterative Gauss Seidel method with sci-py's sparse matrix solver.

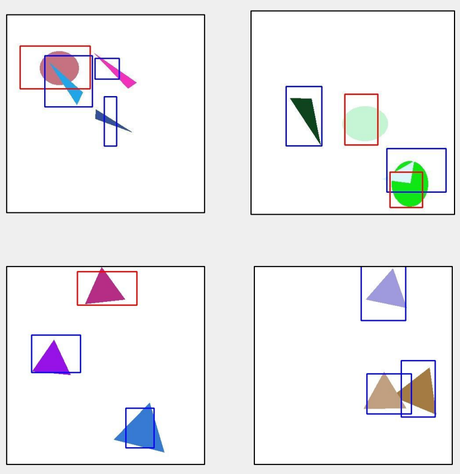
YOLO Object Detection

- The YOLO algorithm is a real-time object detection algorithm
- Predictors are trained to propose bounding boxes and classify objects simultaneously.
- An image into an S x S grid, where each grid cell predicts B bounding boxes. Furthermore, each bounding box consists of a position (x, y) relative to the box, an aspect ratio (w, h) relative to the image, and a confidence defined as Pr(Object) * IOU.
- Finally, each grid cell also predicts C conditional class probabilities defined as $Pr(Class_i \mid Object)$.



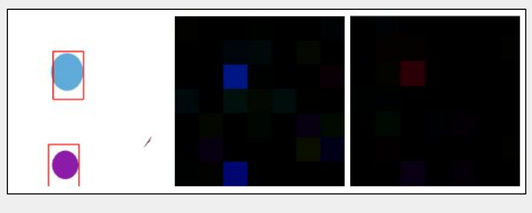
Selected Results

- For testing on the shape framework, we use S = 7, B = 2, and C = 2.
- The network was trained on 1024 images
- We obtained a mAP of 0.7



Predictor Behavior

- The purpose of having each grid cell propose multiple bounding boxes is that each bounding box is supposed to learn different aspect ratios.
- However, perhaps due to the simplicity of the shapes, this was not observed.
- This may explain why it is sufficient that Redmon et. al. only use 2 predictors for a complex dataset like VOC, as each predictor is already very expressive.



Brox Optical Flow

- Brox et. al., choose a combination between brightness constancy, gradient constancy and smoothness for a robust and rotational invariant energy function. The energy can be calculated as:

$$E_{Data}(u,v) = \int_\Omega \Psi\left(|I(\mathbf{x}+\mathbf{w}) - I(\mathbf{x})|^2 + \gamma|\nabla I(\mathbf{x}+\mathbf{w}) - \nabla I(\mathbf{x})|^2\right) \mathbf{dx}$$

$$E_{Smooth}(u,v) = \int_\Omega \Psi\left(|\nabla_3 u|^2 + |\nabla_3 v|^2\right) \mathbf{dx}.$$

$$E(u,v) = E_{Data} + \alpha E_{Smooth}$$

- The vector functions u and v, represent the change in x and the change in y respectively. Hence, u and v must satisfy the euler lagrange equations:

$$\begin{aligned}(\Psi')^k_{Data} &:= \Psi'\Big((I^k_z + I^k_x du^k + I^k_y dv^k)^2 \\ &\quad + \gamma\big((I^k_{xz} + I^k_{xx} du^k + I^k_{xy} dv^k)^2 + (I^k_{yz} + I^k_{xy} du^k + I^k_{yy} dv^k)^2\big)\Big), \\ (\Psi')^k_{Smooth} &:= \Psi'(|\nabla_3(u^k + du^k)|^2 + |\nabla_3(v^k + dv^k)|^2),\end{aligned}$$

- **w** = <u, v> is computed with fixed point iterations.

Fixed Point Iterations

- However, the equation is non-linear, so it is difficult to solve for **w**
- Assume $\mathbf{w}^0$ = <0, 0, 1> and iteratively update **w**
- Fix the spatial functions of $\mathbf{w}^k$ by treating them as constants, and update the temporal functions of $\mathbf{w}^{k+1}$
- Finally, taylor series is used to remove non-linearity as well

$$\begin{aligned}0 = (\Psi')^{k,l}_{Data} \cdot \Big(&I^k_x\big(I^k_z + I^k_x du^{k,l+1} + I^k_y dv^{k,l+1}\big) \\ &+ \gamma I^k_{xx}\big(I^k_{xz} + I^k_{xx} du^{k,l+1} + I^k_{xy} dv^{k,l+1}\big) + \gamma I^k_{xy}\big(I^k_{yz} + I^k_{xy} du^{k,l+1} + I^k_{yy} dv^{k,l+1}\big)\Big) \\ &- \alpha \operatorname{div}\Big((\Psi')^{k,l}_{Smooth} \nabla_3(u^k + du^{k,l+1})\Big)\end{aligned}$$

Selected Results

- Three rectangles with a “collision”

Discussion

Object Detection:

- We did not observe the predictors learning separate aspect ratios
- The second predictor got suppressed by the first one
  - As training progressed, the confidence of the second predictor approached 0
- The network only needed the first predictor to get good results

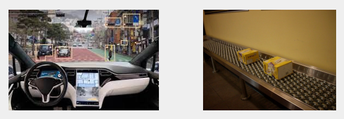
Optical Flow:

- A direct sparse solver was used instead of Gauss-Seidel/SOR iterations
- Performance was slightly slower than the results presented in the Brox paper.
- Due to the white background, the flow field still shows “movement” but this can be attributed to the smoothness factor

Applications

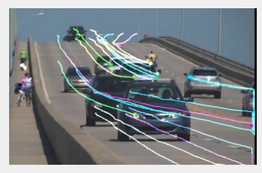
Object Detection:

- Classical example of self-driving cars



Optical Flow:

- Video tracking



Future Research

Object Detection

- Explore how YOLO predictors behave for more complicated datasets like VOC
- State of the art techniques now use segmentation masks instead of bounding boxes

Optical Flow

- Improve speed for optical flow
- Optical flow on compressed images
- Neural network methods for optical flow are hot right now

Selected Citations

- Brox, T., Bruhn, A., Papenberg, N., & Weickert, J. (2004). High accuracy optical flow estimation based on a theory for Warping.
- Fleet, D., & Weiss, Y. (n.d.). Optical flow estimation
- Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection.
- Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards real-time object detection with region proposal networks.