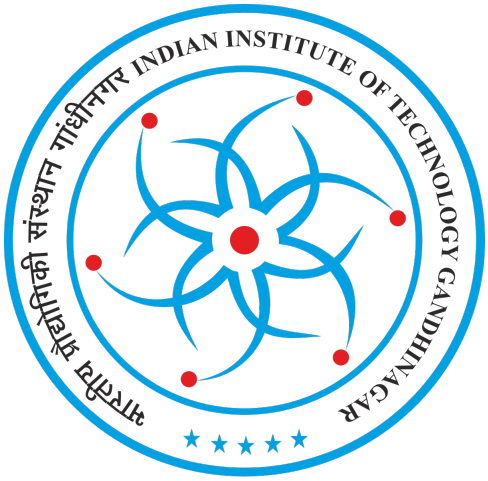

SRIP 2022
SUMMER RESEARCH
INTERNSHIP PROGRAM

Project code: IP0SR000008

Optical Flow Estimation using Graph Convolutional Networks

Under the guidance of:
Prof. Shanmuganathan Raman, IIT Gandhinagar

NAME: Abhay Jain

INSTITUTE: IIT (BHU), Varanasi

DISCIPLINE: Electrical Engineering

INTERN ID: 21146099

Acknowledgement

I would like to thank Prof. Shanmuganathan Raman for giving me an opportunity to work at the CVIG Lab, IIT Gandhinagar and also for his guidance and encouragement. I would also like to thank my mentors, Ashish Tiwari and Prajwal Singh for their valuable inputs. I learned a lot of things from them and the extent of knowledge which I gained was immense.

I also appreciate the opportunity that IIT Gandhinagar provided me to learn, network, and grow through SRIP 2022. This internship has been a unique experience for me and has contributed significantly to my personal and academic growth.

ABHAY JAIN

Optical Flow Estimation using Graph Convolutional Networks

Abhay Jain[1] and Shanmuganathan Raman[2]

[1] Electrical Engineering, Indian Institute of Technology (BHU), Varanasi, India

[2] Indian Institute of Technology Gandhinagar, India

jain.abhayrakesh.cd.eee20@iitbhu.ac.in, shanmuga@iitgn.ac.in

Abstract

Motion is considered to be a vital cue in an image sequence as it helps in revealing the dynamics of scenes by relating spatial image features to temporal changes. The task of motion estimation from a sequence of images remains a challenging and fundamental problem in computer vision. It has garnered considerable attention from researchers around the globe. In all situations except constant illumination, motion estimation is performed by means of optical flow. The purpose of optical flow estimation is to generate a dense 2D real-valued (u,v vector) map of the motion occurring from one video frame to the next. This information can be very useful when trying to solve computer vision problems such as object tracking, action recognition and video object segmentation. Two highly efficient deep learning based approaches for Optical Flow estimation have been proposed in recent years. FlowNet architecture is the first one which was introduced in 2015 as the first CNN approach to predict Optical Flow. The second architecture is that of RAFT - Recurrent All-Pairs Field Transforms which is a composition of CNN and RNN architectures introduced in 2020. Recently, the Graph Convolution Network (GCN) has risen in popularity due to its versatility in solving deeply interconnected real-world problems. It combines the convolutional principle of the more traditional Convolutional Neural Network (CNN) into a graph data structure. In this work, an attempt has been made to propose a GCN based representation learning framework for the task of Optical Flow estimation. With the rapid advancements in graph representation learning to learn robust features over unstructured data, we explore if GCNs can further enhance video understanding, both spatially and temporally. The experiments for optical flow estimation are performed on the MPI Sintel and KITTI datasets by using various GCN models.

Keywords: *Optical Flow, Graph Convolutional Network (GCN), Convolutional Neural Network(CNN), Recurrent Neural Network(RNN)*

1 Introduction

1.1 Background

Optical flow is a fundamental task in video understanding and analysis, aiming to estimate the pixel-wise correspondence between two video frames. It is a long-standing vision problem that remains unsolved. The best systems are limited by difficulties including fast-moving objects, occlusions, motion blur and textureless surfaces. The objective of its estimation is to determine an approximation to the 2D motion field-(a projection of the 3D velocities of surface points onto the imaging plane) from spatio-temporal patterns of image intensity. It has drawn continuous attention from both academia and industry due to its wide applications, e.g., person-identification [1], visual tracking [2] and video inpainting [3]. Recent years have witnessed significant breakthroughs made to push its performance frontier([4]; [5]; [6]; [7]), but it remains challenging due to inherent ambiguity in textures, large displacements, occlusions, motion blur, and non-Lambertian effects.

Traditional optical flow algorithms formulate the dense matching as an energy minimization problem based on feature constancy and spatial smoothness. However, because the hand-designing features and optimization objectives are difficult to cover all scenarios, these approaches are not robust enough to deal with complex motions. As a powerful alternative to traditional methods, deep learning based approaches take the research of optical flow into a new level. Current deep learning methods [4]; [5]; [6] have achieved performance comparable to the best traditional methods while being significantly faster at inference time.

1.2 Statement of Problems

In this work, we investigate the optical flow estimation task using a GCN based framework. A key question is whether we can design an effective GCN architecture that can perform better, train more easily, and generalize well to novel scenes. By doing this, we wish to address the following questions:

- Is the model able to compute the flow field for large/fast motion?
- Is it possible to make the model robust against outliers (occlusions, illumination changes and noise, etc)?
- How to compute optical flow in a fast and accurate manner?

2 Literature Review

2.1 Graph Convolutional Networks

Neural Networks have gained massive success in the last decade. However, early variants of Neural Networks could only be implemented using regular or Euclidean data, while a lot of data in the real world have underlying graph structures which are non-Euclidean. The non-regularity of data structures have led to recent advancements in Graph Neural Networks which were developed by Thomas Kipf and Max Welling [8]. More formally, a graph convolutional network (GCN) is a neural network that operates on graphs. Given a graph G=(V,E) which takes as input:

- A feature description x_i for every node i; summarized in a N×D feature matrix X (N: number of nodes, D: number of input features)
- A representative description of the graph structure in matrix form; typically in the form of an adjacency matrix A.

and produces a node-level output Z (an N×F feature matrix, where F is the number of output features per node). The propagation rule for GCN introduced in [8] is:

$$f(H^{(l)}, A) = \sigma\left(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right),$$

with $\hat{A} = A + I$, where I is the identity matrix and $\hat{D}$ is the diagonal node degree matrix of $\hat{A}$.

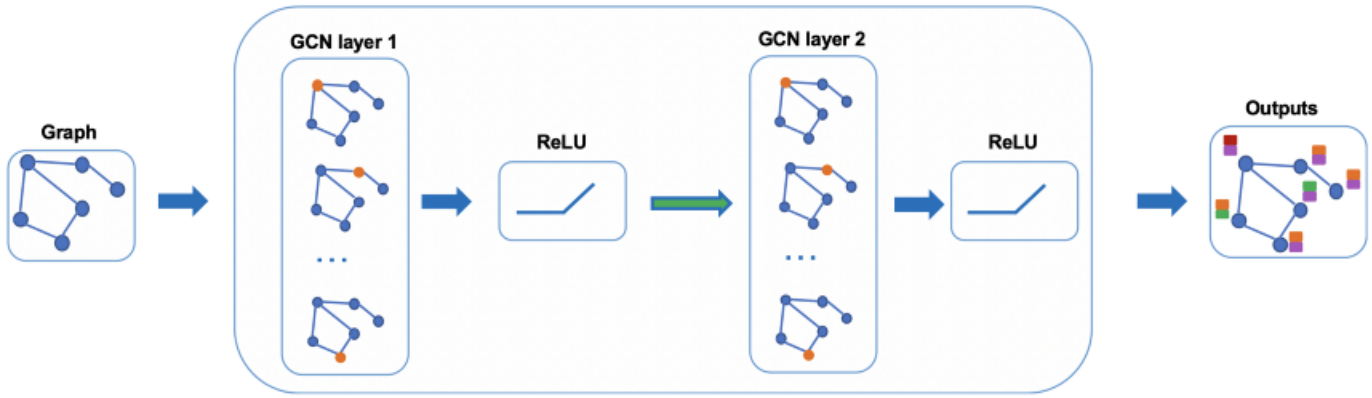

Figure 1: **Example of 2-layer GCN: The output of the first layer is the input of the second layer.**

GNNs in Computer Vision: Using regular CNNs, machines can distinguish and identify objects in images and videos. Although there is still much development needed for machines to have the visual intuition of a human. Yet, GNN architectures have been applied to image classification problems. One of these problems is scene graph generation, in which the model aims to parse an image into a semantic graph that consists of objects and their semantic relationships. However, the number of applications of GNNs in computer vision is still growing. It includes human-object interaction, few-shot image classification, and more.

2.2 Optical Flow

Optical Flow as Energy Minimization: Optical flow has traditionally been treated as an energy minimization problem which imposes a tradeoff between a data term and a regularization term. Horn and Schnuck [9] formulated optical flow as a continuous optimization problem using a variational framework, and were able to estimate a dense flow field by performing gradient steps. TV-L1 [10] replaced the quadratic penalties with an L1 data term and total variation regularization, which allowed for motion discontinuities and was better equipped to handle outliers.

Direct Flow Prediction: Neural networks have been trained to directly predict optical flow between a pair of frames, side-stepping the optimization problem completely. Coarse-to-fine processing has emerged as a popular ingredient in many recent works [11,12,13,14,15].

Iterative Refinement for Optical Flow: Many recent works have used iterative refinement to improve results on optical flow [5,16,17] and related tasks. Ilg et al. [5] applied iterative refinement to optical flow by stacking multiple FlowNetS and FlowNetC modules in series. SpyNet[16], PWC-Net[17] and LiteFlowNet[18] apply iterative refinement using coarse-to-fine pyramids.

Learning to Optimize: Many problems in vision can be formulated as an optimization problem. These works typically use a network to predict the inputs or parameters of the optimization problem, and then train the network weights by back-propogating the gradient through the solver. The RAFT: Recurrent All-Pairs Field [19] for Optical Flow architecture is considered as learning to optimize since the network uses a large number of update blocks to emulate the steps of a first-order optimization algorithm.

Figure 2 illustrates an example of estimated optical flow from a Rubik cube image sequence, in which a cube is rotating in counter-clockwise direction on a turntable. The flow field is given with vector representation and color map of the optical flow field. In vector representation of the optical flow, the motion of a pixel/object between two image frames is represented by an arrow. Therefore, the motion is shown only for Rubik cube.

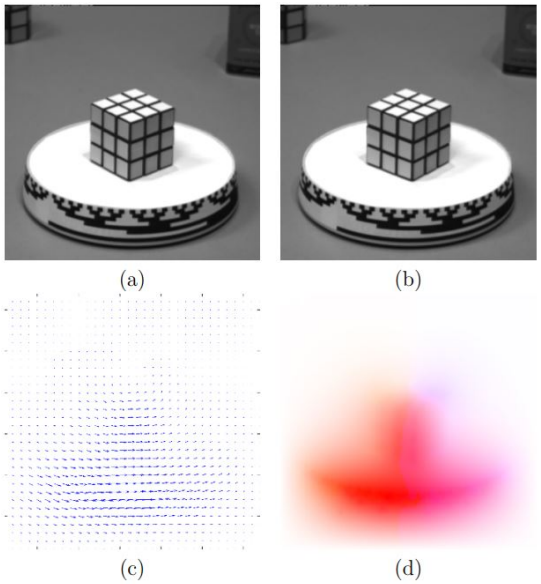

Figure 2: **Optical flow representation for Rubik cube image sequence: (a)Reference image, (b)Target image, (c)Vector form of optical flow field and (d)Color map of the optical flow.**

The color encoding scheme of the optical flow color map is carried out using the source code [20]. This code is based on the theory of color wheel. The figure given below represents the color wheel diagram made from different colors and which is further used to code the color map of the optical flow. In color map of the optical flow, different color represents different directions and homogeneous region represents large displacement. This color map shows the rotation of colors in counter-clockwise direction.

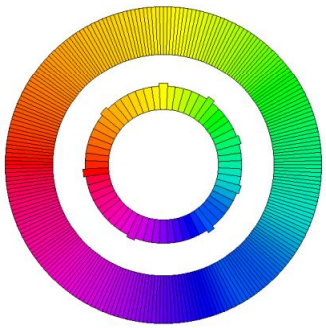

Figure 3: **Color wheel diagram used for a color encoding scheme of the optical flow map**

3 Methodology

An encoder-decoder based Graph Convolutional framework is proposed to estimate the flow information from a pair of consecutive video frames as input. The first step is the graphical representation of a frame. A 64 x 64 image is represented as a graph where the nodes represent the image pixels and edges represent an 8-connected neighbourhood. A graph is constructed where each node (pixel) is connected only to its 8-neighbouring pixels (nodes). The resulting adjacency matrix $A \in R^{64^2 \times 64^2}$.

Next, we construct a the feature matrix $X \in R^{64^2 \times 7}$ where each row represents the respective node features. We obtain the 7-dimensional node feature by using the pixel-positions (2), RGB values (3), and the gradient values (2) for each node. Using position values as initial features helps in consistent mapping of pixel information from graph to grid representation. Further, the gradient values help in better understanding the flow and the variation in the image features.

- **GCN based Architecture for Optical Flow Estimation:**

We construct a six layer GCN based framework for the task of flow estimation. Figure 4 shows the block architecture of the proposed framework. Let (G_1, G_2) be the graph representation of a pair of video frames (I_1, I_2). The encoder consists of three GCN layers to extract features (F_1, F_2) from the two input graphs (G_1, G_2).

Next, we compute the correlation between (F_1, F_2) to obtain the feature matrix $F \in R^{64^2 \times 64^2}$ also called, visual similarity. It is calculated as the inner product of all pairs of node features and gives crucial information about small and large pixel displacements. The feature F is then passed through the decoder consisting of three GCN layers.

We combine the features from different decoder layers (similar to multi-scale feature fusion) to finally obtain the flow map. However, since here we do not alter the scale, combining features from different decoder layers can be considered as hierarchical feature fusion. Experiments are performed on three different GCN models namely GCNII [21], DeepGCN (ResGCN, DenseGCN) [22] and GCN with convolutional ARMA filters [23].

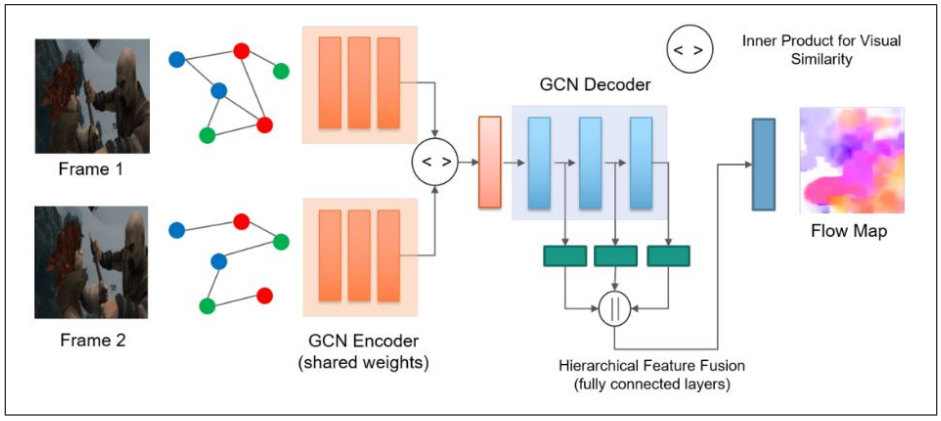

Figure 4: **Architecture of the proposed GCN framework for Optical Flow Estimation**

Model 1 (GCNII): For Model 1, The GCN layers in the above framework are mathematically described by the following equation (as proposed by [21])

$$\mathbf{H}^{(\ell+1)} = \sigma\Big(\big((1-\alpha_\ell)\tilde{\mathbf{P}}\mathbf{H}^{(\ell)} + \alpha_\ell \mathbf{H}^{(0)}\big)\big((1-\beta_\ell)\mathbf{I}_n + \beta_\ell \mathbf{W}^{(\ell)}\big)\Big)$$

Here, $\tilde{P} = \tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}$ is the normalised graph Laplacian matrix. α_1 and β_1 are weights assigned to initial features $H^{(0)}$ and the identity mapping, respectively.

Model 2 (DeepGCNs): A key reason behind the success of CNNs is the ability to design and reliably train very deep CNN models. In contrast, it is not yet clear how to properly train deep GCN architectures. Stacking more layers into a GCN leads to the common vanishing gradient problem.

This means that back-propagating through these networks causes oversmoothing, eventually leading to features of graph vertices converging to the same value. The "DeepGCNs: Can GCNs Go as Deep as CNNs?" paper [22] solves this issue by presenting some novel ways to successfully train very deep GCNs by borrowing concepts from CNNs, specifically residual/dense connections and dilated convolutions, and adapting them to GCN architectures.

ResGCN - ResGCN is constructed by adding dynamic dilated k-NN and residual graph connections to GCN.

DenseGCN - Similarly, DenseGCN is built by adding dynamic dilated k-NN and dense graph connections to the GCN. Dense graph connections are created by concatenating all the intermediate graph representations from previous layers.

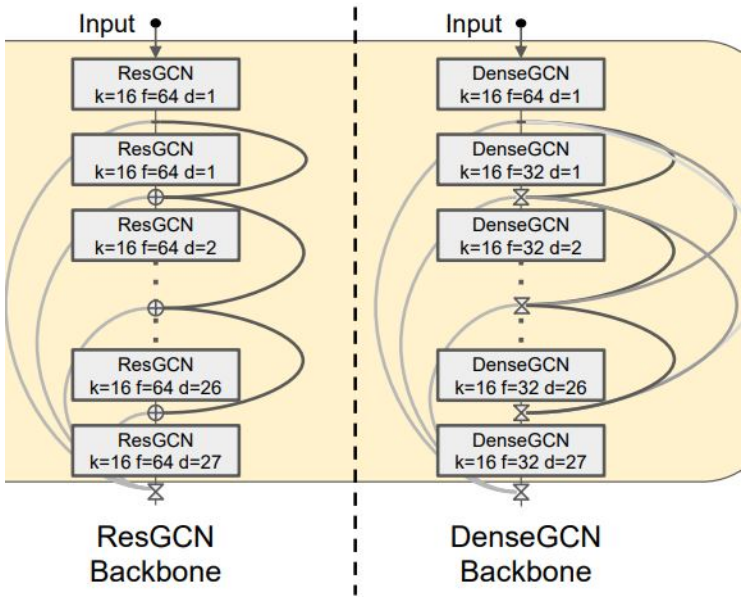

Figure 5: **Architecture of ResGCN and DenseGCN**

Model 3 (GCN with ARMA filter): Popular graph neural networks implement convolution operations on graphs based on polynomial spectral filters. In the paper [23], a novel graph convolutional layer inspired by the auto-regressive moving average (ARMA) filter is proposed that, compared to polynomial ones, provides a more flexible frequency response, is more robust to noise, and better captures the global graph structure.

Model 4 (Capsule Graph Neural Networks; yet to be experimented for the task of Optical Flow Estimation):

The high-quality node embeddings learned from the Graph Neural Networks (GNNs) have been useful in applying to numerous node-based applications and some of them have achieved state-of-the-art (SOTA) performance. However, in the process of applying node embeddings learned from GNNs to generate graph embeddings, the scalar node representation may not suffice to preserve the node/graph properties efficiently, resulting in sub-optimal graph embeddings. Inspired by Capsule Neural Network (CapsNet), the Capsule Graph Neural Network (CapsGNN) [24] was proposed which adopts the concept of capsules to address the weakness in existing GNN-based graph embeddings algorithms. The architecture of the model for graph classification tasks is depicted in Figure 6.

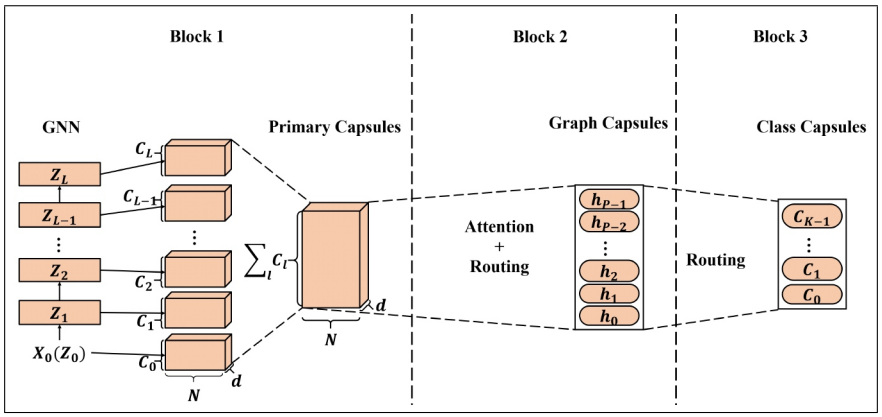

Figure 6: **Architecture of CapsGNN for Graph Classification tasks as proposed by [24]**

The CapsGNN model has been applied successfully to various graph classification tasks however the applications of this model in Computer Vision are too less. Inorder to understand the architectural changes required in CapsGNN for vision related tasks, I implemented the paper [25] (which is aimed at object recognition using CapsGNN) with the help of the pseudo-code(mentioned in the paper) as the code was not available for the same. Capsule Networks, as alternatives to Convolutional Neural Networks, have been proposed to recognize objects from images. Inspired by this, the authors have proposed Graph Capsule Networks for Object Recognition. In this framework, the relationship between the primary capsules is modelled (i.e., part-part relationship) with graphs. Then, the followed graph pooling operations pool relevant object parts from the graphs to make a classification vote.

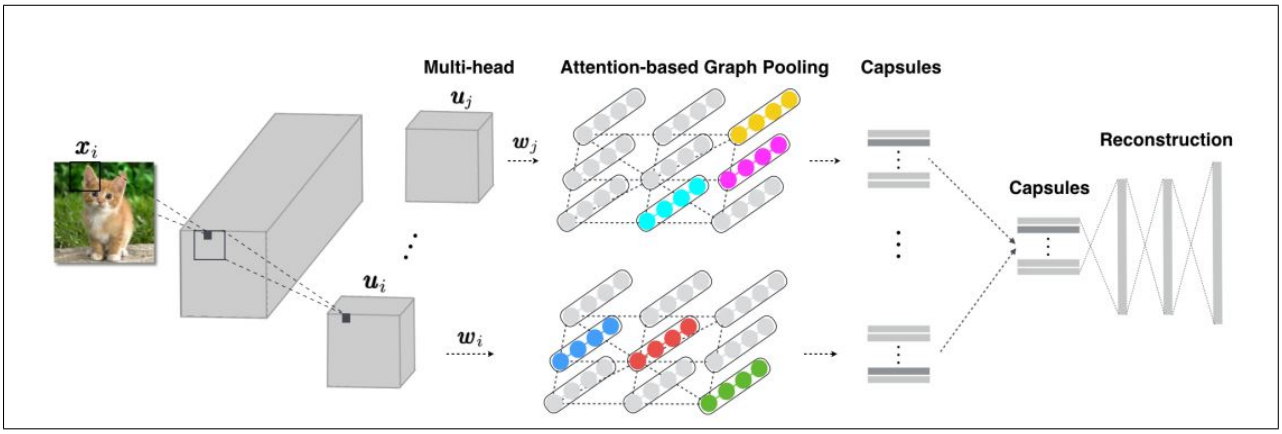

Figure 7: **Illustration of the above model (as proposed by [25]): The extracted primary capsules are transformed and modeled as multiple graphs. The pooling result on each graph (head) corresponds to one vote. The votes on multiple graphs (heads) are averaged to generate the final prediction.**

This model helped me in understanding the framework of CapsGNN for computer vision related tasks. I will try and experiment this model further for our task of Optical Flow Estimation.

4 Experimental Analysis

- **Dataset**

The MPI Sintel dataset obtains ground truth from rendered artificial scenes with special attention to realistic image properties. Two versions are provided: the Final version contains motion blur and atmospheric effects, such as fog, while the Clean version does not include these effects. Sintel is the largest dataset available (1,041 training image pairs for each version) and provides dense ground truth for small and large displacement magnitudes. The important features that dataset contains are long sequences, large motions, specular reflections, motion blur, defocus blur, and atmospheric effects.

The KITTI dataset contains a suite of vision tasks built using an autonomous driving platform. The full benchmark contains many tasks such as stereo, optical flow, visual odometry. It consists of 200 training scenes and 200 test scenes and includes large displacements, but contains only a very special motion type. The ground truth is obtained from real world scenes by simultaneously recording the scenes with a camera and a 3D laser scanner.

4.1 Results obtained for the above mentioned GCN models:

- **Model 1 (GCNII)**

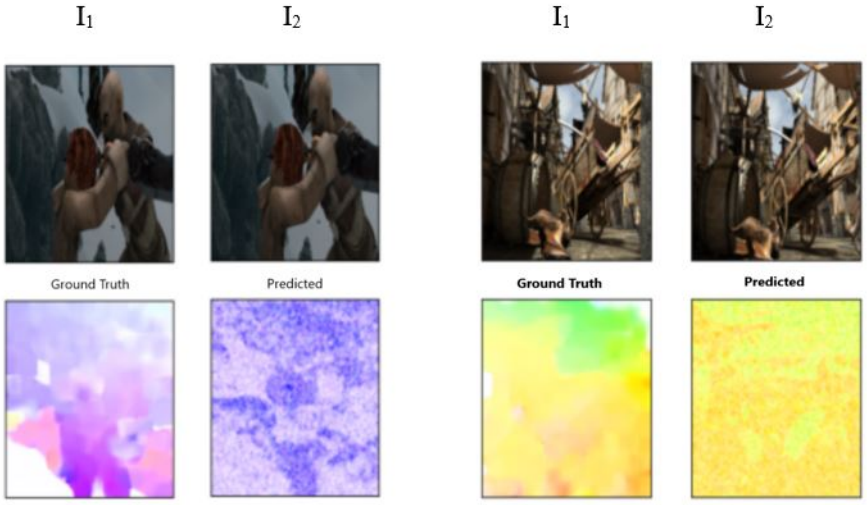

Figure 8a: **Results (Model 1) for flow prediction on MPI Sintel dataset**

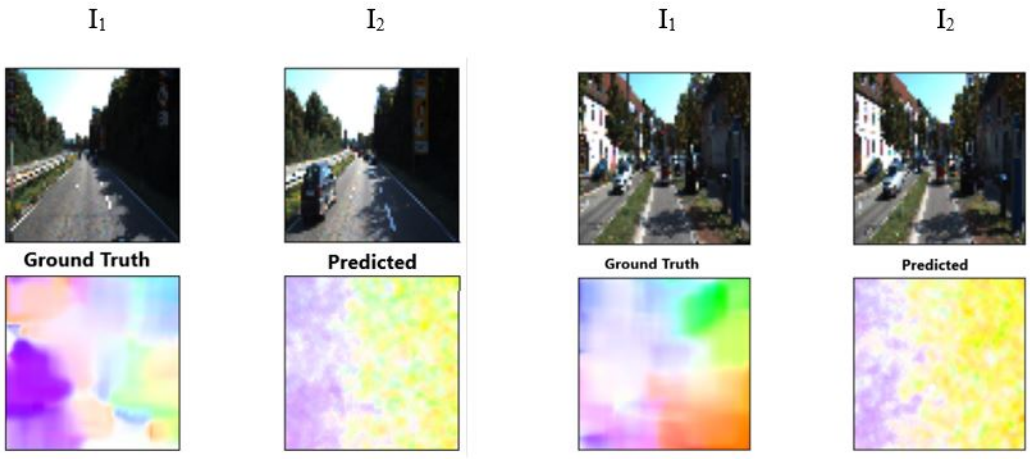

Figure 8b: **Results (Model 1) for flow prediction on KITTI dataset**

- **Model 2 (DeepGCN)**

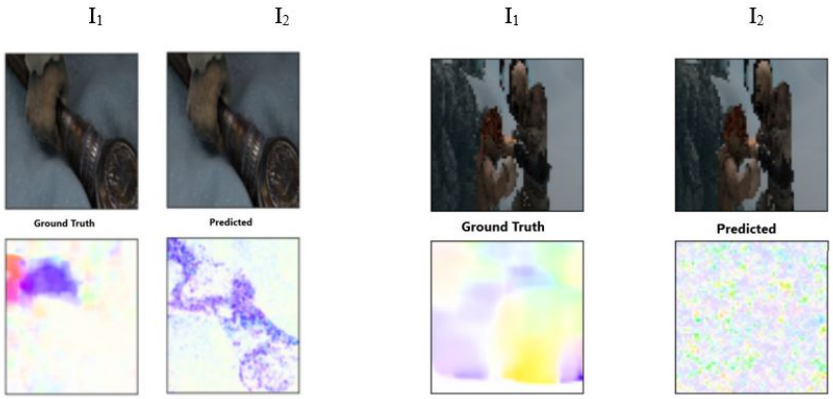

Figure 8a: **Results (Model 2) for flow prediction on MPI Sintel dataset**

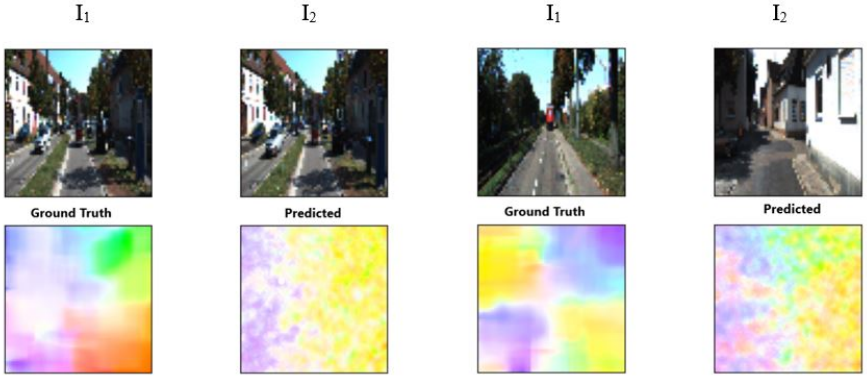

Figure 8b: **Results (Model 2) for flow prediction on KITTI dataset**

- **Model 3 (GCN with Convolutional ARMA filter)**

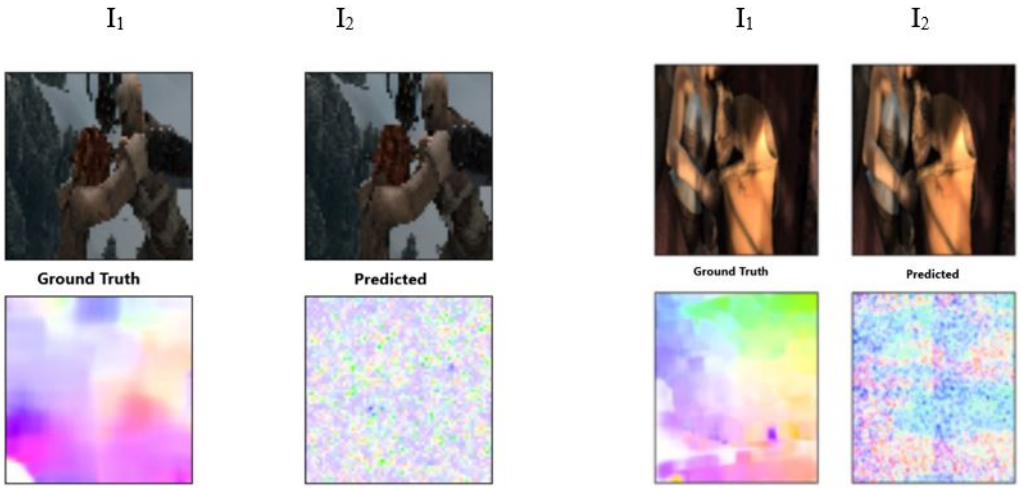

Figure 10a: **Results (Model 3) for flow prediction on MPI Sintel dataset**

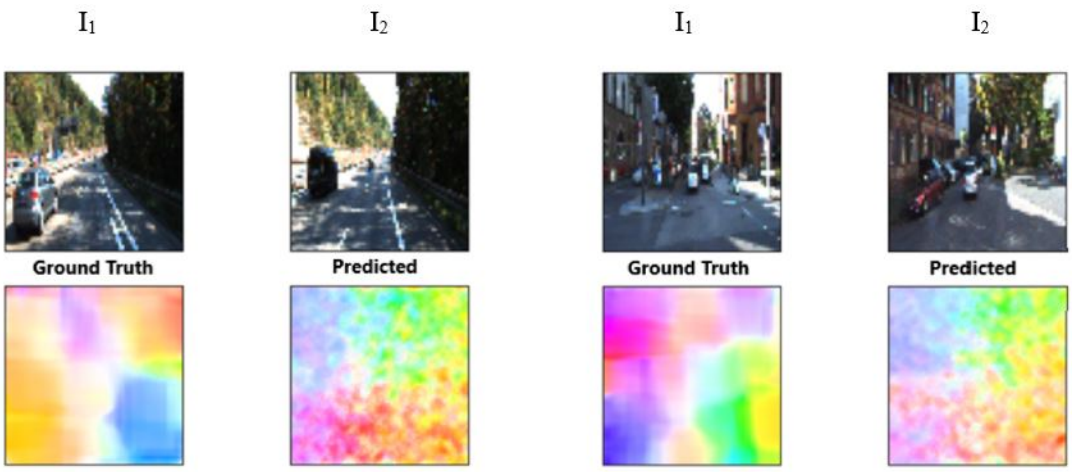

Figure 10b: **Results (Model 3) for flow prediction on KITTI dataset**

The flow maps obtained are sparse in nature, as shown in the above figures. Furthermore, the performance over image pairs with very little optical flow is not that convincing when compared to that on the image pairs with large optical flow magnitude. As a result, the proposed method doesn't outperform the CNN based baseline models for optical flow. However, the proposed GCN formulation is able to learn the semantics from the image pairs (ofcourse not as detailed as CNNs) to an appreciable extent.

5 Challenges and Future Work

The applications of GCNs in computer vision are still growing. In the above GCN frameworks, the performance over image pairs is not so good as compared to state of the art models. Recently, the "Capsule Graph Neural Networks" by Zhang Xinyi, Lihui Chen [24] was introduced which has been applied for mainly graph classification tasks. In Section 3 of this report, I had mentioned about an application of Model 4 (CapsGNN) in vision related tasks i.e for the task of Object Recognition. The concept of capsules was recently invented by Hinton's team and was applied to a wide range of tasks which performed quite successfully. CapsNet, which is designed for extraction of image features was developed based on CNN. However, unlike traditional CNN in which the presence of feature is represented with scalar value in feature maps, the features in CapsNet are represented with capsules (vectors).

Inspired by CapsNet, the capsule mechanism is adopted and fused with GNN in the CapsGNN to generate graph capsules and class capsules on the basis of node capsules which are extracted from GNN. Recently, a CapsNet based architecture, termed FlowCaps [26] was proposed for the task of Optical flow estimation and it has outperformed several state-of-the art models like RAFT [19] and FlowNet [5]. A combination of CapsGNN and CapsNet for the task of optical flow estimation might prove to be beneficial hence in future, I would like to work on developing such an architecture. A novel graph-based approach, called adaptive graph reasoning for optical flow (AGFlow) [28] was also proposed recently, to emphasize the value of scene/context information in optical flow. This framework is entirely based on RAFT but a novel graph based adapter has been introduced for better performance. We can also exploit other attributes from this model to design an efficient framework

6 References

[1] Chen, Z.; Zhou, Z.; Huang, J.; Zhang, P.; and Li, B. 2020. Frame-guided region-aligned representation for video person re-identification. In AAAI.

[2] Vihlman, M.; and Visala, A. 2020. Optical Flow in Deep Visual Tracking. In AAAI.

[3] Xu, R.; Li, X.; Zhou, B.; and Loy, C. C. 2019. Deep FlowGuided Video Inpainting. In CVPR.

[4] Dosovitskiy, A.; Fischer, P.; Ilg, E.; Hausser, P.; Hazirbas, ¨ C.; Golkov, V.; Smagt, P. V. D.; Cremers, D.; and Brox, T. 2015. FlowNet: Learning Optical Flow with Convolutional Networks. In ICCV.

[5] Ilg, E.; Mayer, N.; Saikia, T.; Keuper, M.; Dosovitskiy, A.; and Brox, T. 2017. FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks. In CVPR.

[6] Teed, Z.; and Deng, J. 2020. RAFT: Recurrent All-Pairs Field Transforms for Optical Flow. In ECCV.

[7] Jiang, S.; Lu, Y.; Li, H.; and Hartley, R. 2021b. Learning Optical Flow from a Few Matches. In CVPR.

[8] Kipf, T.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In ICLR.

[9] Horn, B.K., Schunck, B.G.: Determining optical flow. In: Techniques and Applications of Image Understanding. vol. 281, pp. 319–331. International Society for Optics and Photonics (1981)

[10] Zach, C., Pock, T., Bischof, H.: A duality based approach for realtime tv-l 1 optical flow. In: Joint pattern recognition symposium. pp. 214–223. Springer (2007)

[11] Sun, D., Yang, X., Liu, M.Y., Kautz, J.: Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8934–8943 (2018)

[12] Yin, Z., Darrell, T., Yu, F.: Hierarchical discrete distribution decomposition for match density estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6044–6053 (2019)

[13] Hui, T.W., Tang, X., Change Loy, C.: Liteflownet: A lightweight convolutional neural network for optical flow estimation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8981–8989 (2018)

[14] Hui, T.W., Tang, X., Loy, C.C.: A lightweight optical flow cnn–revisiting data fidelity and regularization. arXiv preprint arXiv:1903.07414 (2019)

[15] Zhao, S., Sheng, Y., Dong, Y., Chang, E.I., Xu, Y., et al.: Maskflownet: Asymmetric feature matching with learnable occlusion mask. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6278–6287 (2020)

[16] Ranjan, A., Black, M.J.: Optical flow estimation using a spatial pyramid network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4161–4170 (2017)

[17] Sun, D., Yang, X., Liu, M.Y., Kautz, J.: Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8934–8943 (2018)

[18] Hui, T.W., Tang, X., Change Loy, C.: Liteflownet: A lightweight convolutional neural network for optical flow estimation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8981–8989 (2018)

[19] Teed, Z.; and Deng, J. 2020. RAFT: Recurrent All-Pairs Field Transforms for Optical Flow. In ECCV

[20] Baker, S., Scharstein, D., Lewis, J. P., Roth, S., Black, M. J. and Szeliski, R. A database and evaluation methodology for optical flow. International Journal of Computer Vision

[21] Chen, M., Wei Z., Huang, Z., Ding, B., Li, Y., Simple and Deep Graph Convolutional Networks

[22] Li G., Muller M., Thabet A., Ghanem B., DeepGCNs: Can GCNs Go as Deep as CNNs?

[23] Bianchi F., Grattarola D., Livi L., Alippi C., Graph Neural Networks with Convolutional ARMA Filters

[24] Xinyi Z., Chen L., Capsule Graph Neural Network

[25] Chen Z., Wei X., Wang P., Guo Y., Multi-Label Image Recognition with Graph Convolutional Networks

[26] Jayasundara V., Roy D., Fernando B., FlowCaps: Optical Flow Estimation with Capsule Networks For Action Recognition

[27] Luo A., Yang F., Luo K., Li X., Fan H., Liu S., Learning Optical Flow with Adaptive Graph Reasoning