FlowCaps: Optical Flow Estimation with Capsule Networks For Action Recognition

Vinoj Jayasundara, Debaditya Roy, and Basura Fernando

SCC, IHPC, A*STAR, Singapore.

Abstract

Capsule networks (CapsNets) have recently shown promise to excel in most computer vision tasks, especially pertaining to scene understanding. In this paper, we explore CapsNet's capabilities in optical flow estimation, a task at which convolutional neural networks (CNNs) have already outperformed other approaches. We propose a CapsNet-based architecture, termed FlowCaps, which attempts to a) achieve better correspondence matching via finer-grained, motion-specific, and more-interpretable encoding crucial for optical flow estimation, b) perform better-generalizable optical flow estimation, c) utilize lesser ground truth data, and d) significantly reduce the computational complexity in achieving good performance, in comparison to its CNN-counterparts.

1. Introduction

Optical flow represents apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer and a scene [15]. Given a pair of images from the scene, optical flow estimates the displacement of pixels in spatial domain. Optical flow is important for many applications, including action recognition, motion detection, tracking and autonomous driving. In the recent years, convolutional neural networks (CNNs) have made breakthroughs in a variety of computer vision tasks, including optical flow estimation [6, 24, 16, 32, 34, 36, 2]. For example, FlowNet is an end-to-end trainable CNN to solve the optical flow estimation problem in a data-driven, supervised fashion, which outperforms the conventional curated-feature driven models such as [25]. Ideally, deep optical flow estimation methods should be equivariant which allows us to obtain feature representation equivalent to geometric changes in the image space. This would allows us to obtain accurate optical flow estimation by measuring feature displacements in functional form using deep neural networks. Despite the success of CNNs and CNN-based optical flow estimation, they suffer from the issue of invariance to certain geometric attributes such as translation and affine changes. On the other-hand capsule networks (CapsNets) [14, 28] marked a milestone by identifying and attempting to resolve several key limitations of CNNs, such as the inability to understand spatial relationships between features, being invariant rather than equivariant, inept routing of data between the layers, among others. Therefore in this paper we exploit capsule networks for optical flow estimation task, and to explore the aptness and potential gains caused by the resulting optical flows in auxiliary tasks such as action recognition.

Precise optical flow estimation requires pixel-wise localization as well as correspondence matching between frames. However, raw pixel intensities from a pair-of-frames carry sparse motion-related information useful for optical flow estimation, often cluttered with non-motion related information. Hence, a key sub-task of an optical flow estimator is to successfully untangle motion-related information from raw pixel intensities. Yet, this sub-task is specially challenging for a CNN, since they learn primarily invariant encoding comprising of high level entities, and often lose other useful internal information pertaining to the pose, orientation, whole-part relationships, and other physical properties of such entities, along with the spatial relationships between them. Furthermore, discarding information that are not useful to the task at hand at a lower level, prior to passing them to higher levels is crucial to the untangling process. Yet, CNNs fail to filter out such unnecessary information since they routes data between layers using pooling operators, especially during the initial training iterations. Hence, we argue that the optical flow estimation task will heavily benefit from a more comprehensive and selective encoding mechanism provided by capsule networks.

Capsule networks excel at comprehensively encoding the physical properties of the entities present in the inputs within their instantiation parameter vectors, while learning the part-whole spatial relationships between such entities.

Studies have shown that the physical properties captured by the capsules are often relevant to the task at hand, especially the properties corresponding to the instantiation parameters with the highest variances, and that the encoding learnt by capsules are highly interpretable, by means of a post-training perturbation analysis [19]. The same observation can be extended in our case to assume that the representation learnt by a capsule encoder will comprise motion-specific properties useful for optical flow estimation. Hence, similar entities in the input images receive similar encoding with finer-grained representations than CNNs, allowing the correspondence-matching to be more convenient and precise. Furthermore, the dynamic routing algorithm [28] deployed in capsule networks achieves coincidence filtering, which arguably aids the untangling process. Higher level capsules represent complex motion-related entities with higher degree of freedom, and dynamic routing ensures that lower level entities that have little agreement with these (non-motion related entities) are effectively cut-off from the forward propagation. Hence, we hypothesize that the use of a capsule encoder will aid the optical flow estimation task by providing finer-grained, motion-specific and more-interpretable encoding, in comparison to its CNN counterpart.

Optical flow estimation is an object or class agnostic task. The specific identities and the categories of the objects and the actors are not attributed to the task, only where and what kind of motion take place instead. Hence, it is intuitive that, if the meta-level conditions (such as the presence of camera motion, range of displacement, and etc.) do not drastically change, optical flow estimation models should generalize beyond the data that they are trained for. To further explore this property, among other reasons, we consider action recognition as an auxiliary task. More specifically, we compare the capabilities of CNN and capsule networks in estimating class-agnostic optical flows, and generalizing to other classes when trained on a subset of action classes. CNNs are generally translation invariant, and not invariant to other transforms such as the orientation changes. Hence, they require a lot of training data with ample variations to learn to handle such transforms, resulting in reduced generalization capabilities. In contrast, capsule networks are equivariant, where lower level capsules exhibit place-coded equivariance and higher level capsules exhibit rate-coded equivariance [28]. For instance, CNNs learn rotational invariance by training on a large number augmented images, whereas capsule networks learn to encode rotation in their instantiation parameters without observing many such augmentations. Subsequently, capsule networks are able to successfully encode rotation, even for an image outside the training domain. Hence, we argue that capsule networks will better-generalize to unseen action classes for the optical flow estimation, and require comparatively lesser ground truth data with fewer augmentations to achieve similar performances as CNNs.

In addition, capsule encoders provide a low-dimensional concise representation in comparison to shallow convolutional feature maps, and they undertake a significant portion of the burden of untangling motion-related information. Hence, it reduces the workload from network that generates optical flow image known as expanding network. Simultaneously, a capsule encoder itself has less number of trainable parameters than its direct CNN-counterpart, as capsule networks group neurons together yielding in a fewer number of connections between layers. Hence, a capsule encoder contributes to drastically reducing the computational complexity of the overall.

Estimated optical flows have a wide-utility in a range of computer vision tasks, and action recognition is one such task [30]. It is well known that motion stream obtained via optical flow is complimentary to the spatial stream. As a downstream task, we experiment with action recognition using the estimated optical flows from our models. We investigate two key approaches for this task, the standard frame-wise approach [10] as well as a segment-wise approach which considers a set of consecutive frames together for optical flow estimation (in contrast to two consecutive frames in the frame-wise approach), in an attempt to benefit from the additional contextual information in the segments. We demonstrate that segment-wise optical flow estimation with our model is more accurate and obtains better action recognition results. To this end, we propose a capsule networks-based architecture for optical flow prediction and activity recognition, leveraging on the dynamic routing algorithm [28]. More specifically, we make the following contributions in this paper.

First, we propose a capsule networks based architecture, termed FlowCaps, to achieve better optical flow estimation than its convolutional counterpart. To the best of our knowledge, this is the first attempt to investigate the use of capsules for this task. Furthermore, we utilize the estimated optical flows for action recognition, and propose a modified loss function that improve upon the existing EPE loss.

Second, we evaluate the performance of FlowCaps model on several datasets where we outperform other baselines in both optical flow estimation and action recognition while being less computationally complex. Furthermore, we investigate the capabilities of FlowCaps in terms of out-of-domain generalization and training with only a few samples, in comparison to baselines.

2. Background and Related Works

The concept of grouping neurons to form a capsule was first proposed by Hinton *et al.* in [14] and extended by Sabour *et al.* in [28] introducing the dynamic routing algorithm to route sets of capsules between layers. These mod-

els are primarily used for image classification, image parameterization and image reconstruction. Capsule networks have proven that they excel at various computer vision tasks throughout the literature. CapsGan[18] utilized capsule networks as a discriminator which produced visually better results than conventional CNN-based GANs. Moreover, SegCaps[21] implemented a capsule network based architecture for image segmentation and was able to achieve state-of-the-art results on datasets such as LUNA16. Further, Zhao *et al.*[35] employed capsule networks to classify, reconstruct and perform part-based segmentation on sparse 3D point clouds. Extending capsule networks into video analysis, Duarte *et al.* [7] introduced VideoCapsuleNet which consists of convolutional capsule layers and capsule pooling layers in order to facilitate action recognition. Our method is drastically different from these models as we use capsule network encoder to obtain a representation suitable for optical flow estimation and then use a so called expanding network to generate optical flow images. We also modified the capsule network architecture to avoid issues related to squashing function and make architectural changes to cater for optical flow estimation.

Deep nets based optical flow estimation has been studied in FlowNet, FlowNet 2.0, SpyNet and LiteFlowNet. [6, 17, 24, 16]. LiteFlowNet is composed of two compact sub-networks that are specialized in pyramidal feature extraction and optical flow estimation [16]. This method is able to extract features faster compared to [6]. Similarly, SPyNet [24] also uses spatial pyramids similar to [16] with a compact network. Spatial pyramids are used to make sure that the representation captures some global spatial information. However, our method is able to preserve physical properties in the image space including the spatial structure of entities due to properties of capsule networks. Therefore, we do not need to use a multi-scale approach. Some methods also use additional tools such as external edge detectors or image patch-based correlations in estimating optical flow. Author in [37] interpolates third-party sparse flows using a off-the-shelf edge detector. DeepFlow [33] uses convolution and pooling operations similar to traditional CNNs, however the filter weights are non-trainable image patches. In-fact, similar to FlowNet, DeepFlow also uses correlation. EpicFlow [25] uses externally matched flows as initialization and then performs interpolation. Similarly, Im2Flow [10] and Selflow [22] are related to us. However, to the best of our knowledge, we are the first to use capsule network-based architecture for optical flows estimation which in principle is a better choice than traditional CNNs for this task.

Action recognition methods have benefited a lot from optical flow [30] and other methods use dense optical-flow obtained by motion hallucination for action recognition [10]. Some methods such as Dynamic Images [4, 3] generate motion images for action recognition using rank pooling [9, 8]. Motion images are even used for tasks such as still image action recognition [13] and action anticipation [26]. In this work we also use action recognition as the primary application of optical flow estimation, nevertheless, action recognition is not the primary focus of this paper.

3. FlowCaps: Network Architecture

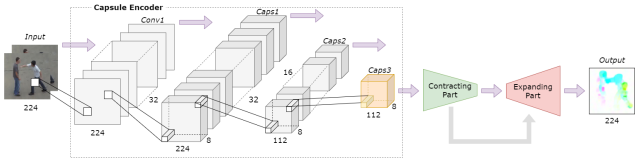


Figure 1. The FlowCaps architecture. The input is fed to the capsule encoder, which passes a concise representation of the input to the subsequent contracting and expanding parts, for optical flow estimation.

Authors of [6] proposed the first end-to-end trainable CNN architecture termed FlowNetS for optical flow estimation. Motivated by the benefits of both FlowNetS and capsule networks, in this work we propose a new model for optical flow estimation by borrowing concepts from both FlowNetS and capsule networks. We call our model FlowCaps-S. Given a pair of RGB images, FlowCaps-S estimate optical flows using the architecture described in section 3.1. If the input is a video, then we show how to use generated optical flow to perform action recognition in section 3.3.

3.1. FlowCaps-S Architecture

In FlowNet, the network learns the optical flow estimation task by applying convolutional filters on the raw pixel values of the considered image pairs. However, extraction of motion information becomes more difficult, especially when the datasets have small realistic displacements. We hypothesize that the use of a capsule encoder as illustrated in Fig. 1 instead of a shallow convolutional encoder prior to the contracting part similar to FlowNet, learns a finer-grained, concise, and more interpretable representation of the physical properties attributed to motion, by eliminating the information unrelated for motion via dynamic routing [28]. Furthermore, our FlowCaps-S benefit from equivariant properties as outlined in the introduction. Now we describe the details of our FlowCaps-S model.

As illustrated in Fig. 1, the proposed capsule encoder consists of a convolutional layer (*Conv1*) having 32 kernels of size 7×7 and a Leaky ReLU activation, followed by three convolutional capsule layers coined *Caps1*, *Caps2* and *Caps3*. The *Caps1* layer consists of 32 channels of 8-dimensional capsules, which will be dynamically routed to each of the 16 channels of 8-dimensional capsules in the

Caps2 layer. We adopt the dynamic routing algorithm proposed in [28], with the exception of squashing of capsule output vectors. The squash function is used to ensure that the length of each capsule output is kept between $0-1$, as the length represents the probability of existence. Albeit existence of motion is crucial for optical flow prediction, we do not utilize the probabilities in a mathematical sense (for instance as in classification), and hence do not require the output lengths to be kept between $0-1$. On the other hand, squashing high dimensional vectors leads to issues such as extremely small individual values and vanishing gradients. Hence, we do not utilize the squash function. Subsequently, the representation is further projected down spatially as well as feature-wise, via dynamic routing to a single channel of 8-dimensional capsules in the *Caps3* layer. The decision to keep the output of the capsule encoder to one channel stems from the capsule representational assumption. Hence, at each location in the input image, there is at most one instance of the type of entity that a capsule represents. This allows us to have a single capsule channel in our representation. This is useful for optical flow estimation task.

Let $\Psi^l \in \mathbb{R}^{(h_1 \times w_1 \times c_1 \times n_1)}$ and $\Phi^l \in \mathbb{R}^{(h_2 \times w_2 \times c_2 \times n_2)}$ be the input and output of the capsule layer l, where h, w are the spatial dimensions, and c, n are the number of channels and the dimensionality of each capsule respectively. Here, $\Phi^{l-1} \equiv \Psi^l$, and ψ_i^l is the $i^{th}(i \in [1, h_1 \times w_1 \times c_1])$ capsule in layer $(l-1)$ and ϕ_j^l is the $j^{th}(j \in [1, h_2 \times w_2 \times c_2])$ capsule in the layer l. First, Ψ^l is reshaped in to $(h_1, w_1, c_1 \times n_1)$ to prepare the channels for the convolution operation, and subsequently convolved with $(c_2 \times n_2)$ filters producing an output of the shape $(h_2, w_2, c_2 \times n_2)$, which is then reshaped to (h_2, w_2, c_2, n_2). Subsequently, each ψ_i^l is routed to each ϕ_j^l dynamically based on their agreement $a_{ij} = \widehat{\psi}_i^l \cdot \phi_j^l$, where $\widehat{\psi}_i^l = W_{ij}\psi_i^l$ and W_{ij} is the trainable transformation matrix which projects ψ_i^l from its native space to the higher dimensional space of ϕ_j^l.

The reduced representation Ψ^3 is then fed to the contracting part followed by the expanding path, which are simplified versions of those proposed in FlowNet. The simplified contracting part comprises seven convolution blocks with batch normalization and Leaky ReLU activation. Downsampling by a factor of 2 occurs every other block, starting from the second, with hyperparameters as illustrated by Fig. 2. The resultant feature map is passed on to the expanding path, which comprises four blocks of upsampling. Each block constitutes a deconvolution layer with Leaky ReLU activation which upsamples the feature maps by a factor of 2, followed by a concatenation with the corresponding block skip connected from the contracting part. Further, we use skip connections between the contracting and expanding parts to nourish the information flow and provide lower-level entity information to the deconvolutional layers, similar to FlowNet.

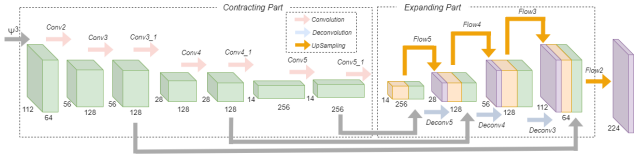


Figure 2. The FlowCaps contracting and the expanding parts. The output of the capsule encoder, Ψ^3, is fed in to the contracting part, followed by the expanding part, which estimates the optical flows.

3.2. Improvements to the Loss Function

The loss function that is used in the state-of-the-art optical flow estimation approaches, endpoint error (EPE), sums the L_2 norms of the difference between the individual components of the ground truth and estimated flow fields. We identify two key issues of using EPE loss in optical flow estimation. First, EPE only considers the magnitude component of the vector field in its calculations, whereas the angle component is omitted. Yet, the angle component carries important information helpful for the optical flow estimation task. Second, the L_2 norm is highly susceptible to outliers with higher values, even a few can have a significant impact on the loss value. In an attempt to alleviate these key issues, we propose the following loss function,

$$L = L_{mag} + \alpha L_{ang} \tag{1}$$

$$L_{mag} = \frac{1}{N}\sum_{i=1}^{N} log(ln(\frac{e^{(u_i^p - u_i^t)} + e^{(u_i^t - u_i^p)}}{2}) + ln(\frac{e^{(v_i^p - v_i^t)} + e^{(v_i^t - v_i^p)}}{2})) \tag{2}$$

$$L_{ang} = \frac{1}{N}\sum_{i=1}^{N}(1 - \frac{u_i^p u_i^t + v_i^p v_i^t}{\sqrt{(u_i^p)^2 + (v_i^p)^2}\sqrt{(u_i^t)^2 + (v_i^t)^2}})||\mathbf{T_i}||_2 \tag{3}$$

where α is an empirically determined constant, N is the mini-batch size, $u_i^p, u_i^t, v_i^p, v_i^t$ are the respective u, v components of the estimated (p) and ground truth (t) optical flows, and $||\mathbf{T_i}||_2 = \sqrt{(u_i^t)^2 + (v_i^t)^2}$. We propose the log-cosh loss function for the magnitude loss L_{mag}, denoted in equation 2, since it is more robust to outliers while behaving similar to the L_2 loss. Furthermore, we propose a variation of the cosine similarity for the angular loss, which adaptively scales the loss with respect to the magnitude of the ground truth optical flows. The scaling aids to alleviate the issue of undefined loss values and gradients when both ground truth and the predicted fields are zero vectors.

3.3. Optical flow estimation for activity recognition

In this study, we attempt two fundamental computer vision tasks, namely, optical flow estimation followed by activity recognition. To this end, we consider two different

approaches based on the number of consecutive frames (k) considered for prediction at a time. First, frame-wise prediction focuses on estimating optical flows from only a pair of consecutive frames ($k = 2$), similar to [6]. Subsequently, the estimated optical flows are utilized in action recognition, producing an action label per the said pair of images. Second, segment-wise prediction focuses on simultaneously estimating the optical flows for a whole segment of consecutive frames ($k > 2$). Similarly, the action recognition is performed on the set of estimated optical flows, producing an action label per the said segment.

We hypothesize that in the datasets where motion information are predominant, action classification performed with optical flows achieves similar performance as with original rgb frames, yet, requires shallower models that are much faster. In the case of datasets where static information are also significant, motion information derived from the optical flows can be combined with the static information to achieve similar performance, similar to [30].

3.4. Frame-wise Prediction

The input to the frame-wise prediction network $\mathbf{X_{frm}} \in \mathbb{R}^{(H \times W \times 2C)}$ is stacked along the channel dimension, where H, W denote the spatial dimensions and C denotes the number of channels per frame. $\mathbf{X_{frm}}$ is fed to the network and the estimated optical flows $\hat{\mathbf{Y}}_{\mathbf{frm}}$ are compared against the ground truth $\mathbf{Y_{frm}} \in \mathbb{R}^{(H \times W \times 2)}$. Note that optical flow images have two channels, the flows in x and y directions. Subsequently, the action recognition is performed with a shallow CNN using predicted optical flow images $\hat{\mathbf{Y}}_{\mathbf{frm}}$ as the input. Our simple optical flow classification network consists of five blocks of convolutional layers and a maxpooling layer each, followed by two fully connected layers. The convolutional layers each have 3×3 filters with ReLU activation, whereas the fully connected layers have 32 and κ units with ReLU and softmax activation respectively, where κ is the number of action classes. We train this network from scratch.

3.5. Segment-wise Prediction

Typically, the optical flow is estimated using consecutive pair of frames, however, models can benefit from additional contextual information. One solution is to predict the optical flow for a given video segment consisting of more than two frames. To be precise, our segment-wise prediction considers k number of frames to be stacked together as the input $\mathbf{X_{seg}} \in \mathbb{R}^{(k \times H \times W \times C)}$. We modify FlowNetS and FlowCaps-S models to handle segment-wise optical flow prediction using 3D convolutions. Specifically, we adopt concepts in I3D [12] and DeepCaps [23]. The modified FlowNetS-3D and FlowCaps-S-3D models estimate the optical flows $\hat{\mathbf{Y}}_{\mathbf{seg}}$ which are compared against the ground truth optical flow $\mathbf{Y_{seg}} \in \mathbb{R}^{(H \times W \times 2)}$ corresponding to the middle two frames of the segment. Subsequently, action recognition is performed with $\hat{\mathbf{Y}}_{\mathbf{seg}}$ similar to Section 3.4.

4. Experiments and Results

In this section we evaluate the validity of our method on several optical flow estimation and video action recognition datasets. Following prior deep optical flow estimation methods [17, 16], we use Sintel [5] and KITTI15 [11] benchmarks for evaluation. We also use the UCF101 [31], UTI [27] KTH [29] and JHMDB [20] datasets for action recognition related experiments. More specifically, for optical flow estimation, we extract frames from the videos and stack k ($k = 2$ for frame-wise, and $k > 2$ for segment-wise) consecutive frames together as the input to the network. However, initial experiments revealed that consecutive frames directly extracted from the videos contained little motion information, yielding trivial optical flows. As a solution, we extracted I-frames and P-frames, corresponding to the keyframes of the video in order to create the following datasets.

KTH I-Frames: All the 6 action classes in the KTH dataset were used for training. After I-frames extraction, 19 videos with only one I-frame were removed, yielding 5,811 samples which are randomly split at a 8:2 ratio for training and testing respectively.

Sub UCF I-Frames: We use the following five classes from the UCF 101 dataset: *Rowing, BenchPress, CleanAndJerk, HulaHoop*, and *Lunges*, selected at random based on the availability of sufficient (more than 5 per video) I-frames, amount of movement and the presence of camera motion. Extraction of I-frames on these five classes yielded in 10,262 optical flow samples which are randomly split at a 8:2 ratio for training and validation respectively. Subsequently, we use I-frames extracted from the rest of the UCF-101 classes as the testing set for out-of-domain generalization.

UTI P-Frames: All the 6 action classes in the UTI dataset were used for training. However, the extraction of I-frames yielded only 1 frame per video. Hence, we decided to use P-frames instead, which yielded in 2,748 optical flow samples, which are split according to the predefined groups as in [27] for training and testing.

Implementation details: We used PyTorch for the development of FlowCaps. All the optical flow estimation models and the action recognition models were trained on GTX-2080Ti and GTX-1080Ti GPUs for 1000 epochs and 50 epochs respectively, using the Adam optimizer with the learning rate set to 0.001. For the models trained on the above datasets, FlowNetS [17] has 38.68 million trainable parameters, whereas the proposed FlowCaps-S has only 2.39 million trainable parameters. This is a drastic reduction in the computational complexity with a significant mar-

Model		Params (M)	Sintel clean	Sintel final	KITTI15
Conventional	EpicFlow [25]	-	2.27	3.56	9.27
	FlowFields [1]	-	**1.86**	3.06	8.33
Heavyweight CNN	FlowNetS [6]	38.68	4.50	5.45	-
	FlowNet2 [17]	162.49	2.02	3.54	10.08
Lightweight CNN	LiteFlowNet [16]	5.37	2.48	4.04	10.39
	SPyNet [24]	1.20	4.12	5.57	-
	Ours	2.39	2.13	**2.51**	**7.83**

Table 1. Comparison of frame-wise training EPE values across different approaches for optical flow estimation datasets.

Model	UCF I-Frames		UTI P-Frames		KTH I-Frames		JHMDB	
	test epe	action	test epe	action	test epe	action	test epe	action
GT	-	79.4%	-	81.37%	-	68.90%	-	51.49%
FlowNetS	1.53	55.58%	0.44	84.12%	1.19	61.30%	0.49	44.03%
LiteFlowNet	-	-	-	83.17%	-	59.79%	-	40.30%
SPyNet	**1.37**	**65.78%**	0.42	87.66%	0.95	64.30%	0.44	42.54%
Ours	1.49	64.49%	0.39	86.02%	1.10	65.00%	**0.40**	**48.51%**
Ours - Mod Loss*	1.41	-	0.35	-	1.04	-	0.26	-
Ours - Segment	1.40	65.16%	**0.37**	**88.34%**	**0.93**	**72.50%**	0.71	41.90%

Table 2. Comparison of frame-wise testing EPE values and action recognition performances by different approaches on the benchmark action datasets. *Training utilizing the modified loss function proposed in eq. 1, and testing using the EPE loss for comparison.

Model	KTH-I Frames		Sub UCF-I Frames		UTI-P Frames	
	Optical flow estimation performance in EPE					
	Frame	Seg.	Frame	Seg.	Frame	Seg.
FlowNetS	1.1934	1.1355	2.3149	2.3079	0.4426	0.4265
FlowCaps-S	1.1033	**0.9384**	2.2037	**2.1930**	0.3806	**0.3672**
	Action classification performance					
FlowNetS	61.30%	66.30%	85.50%	89.70%	84.12%	83.08%
FlowCaps-S	65.00%	**72.50%**	91.20%	**92.30%**	**86.02%**	85.93%
GT	68.90%		92.60%		81.37%	

Table 3. The frame-wise and segment-wise testing EPE values and classification performance achieved by FlowNetS and FlowCaps-S models on the KTH I-frames, Sub UCF I-frames, and UTI P-frames datasets.

gin of 94% by FlowCaps-S, while surpassing the performance of FlowNetS. The reduction in computational complexity can be directly attributed to the hypothesized concise representation achieved by the capsule encoder.

4.1. Evaluating optical flow estimation

In this section we compare our FlowCaps-S model with other state of the art methods in the literature [25, 1, 6, 17, 16, 24]. We train all models in Singtel clean dataset and then evaluate the performance on other dataset using the same model. For comparisons, we may classify prior methods into three categories; "conventional", "heavyweight" and "lightweight" CNN. Our method also falls under lightweight CNN category. Results are reported in Table 1. From the results, we can conclude that our method performs the best in lightweight category and only FlowFields [1] method outperforms our results on Sintel clean dataset. Our method outperforms recent methods such as LiteFlowNet [16] across all compared datasets and our method has good out of domain generalization performance. Our method attains good results due to the suitable properties of capsule networks for optical flow estimation task such as equivariance. We conclude on conventional datasets, our method is able to outperform most of the other methods by a considerable margin for optical flow estimation utilizing relatively small amount of parameters.

4.2. Evaluating on action recognition

In this section we compare our FlowCaps model with other methods using four action recognition benchmarks for optical flow estimation and action recognition. Results are reported in Table 2. Overall our method obtains better results than other recent methods on several datasets. JHMDB is a dataset having lot of motion and it benefits from motion stream. While the ground truth optical flow obtains a classification accuracy of 51.49% our FlowCaps model obtains 48.51 outperforming all other state-of-the-art methods such as SPyNet. Interestingly, on this dataset we obtain the best EPE score of 0.40 while SpyNet obtains only 0.44. On UCF-I frames the ground truth optical flow obtains 79.4% while our method obtains the only of 60.05% where state-of-the-art SPyNet obtains better results than us. We conclude that out model performs well on both traditional optical flow estimation benchmarks as well as on action recognition datasets.

Furthermore, we investigate on the effect of the modi-

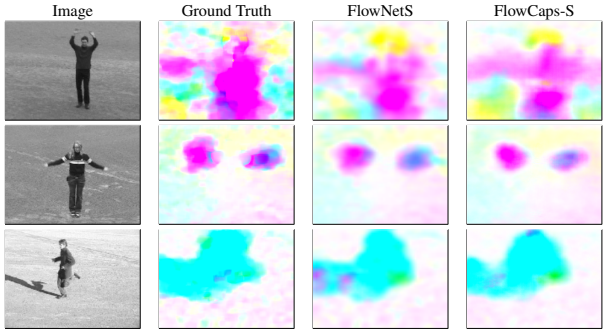


Figure 3. The optical flow estimation results on the KTH I-Frames dataset.

fications proposed to the loss function in Equation 1, with α empirically set to $0.1 - 0.2$. For a fair comparison, we train using the modified loss, while testing with the conventional EPE loss. We obtain significant improvements in optical flow estimation for all four datasets, as reported in Table 2 (Ours - Mod Loss*), establishing that the proposed modifications to the loss function are effective.

4.3. Evaluating the impact of segment-wise and frame-wise model

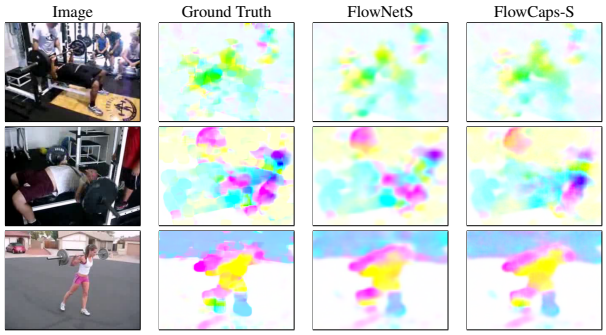


Figure 4. The optical flow estimation results on the UCF I-Frames dataset.

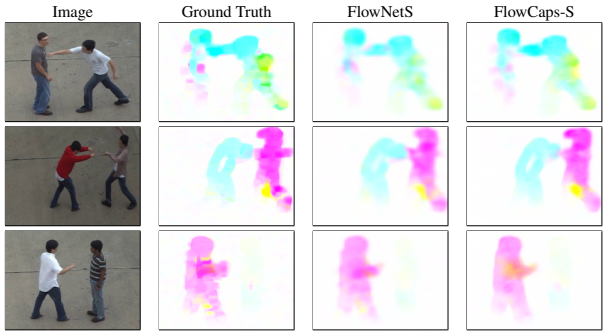


Figure 5. The optical flow estimation results on the UTI P-Frames dataset.

Table 3 compare our results to that of FlowNetS obtained for frame-wise and segment-wise optical flow estimation. On the KTH I-frames dataset, the proposed FlowCaps-S model outperformed the FlowNetS model by a significant margin of 7.55%, by achieving an average testing EPE value of 1.1033. The relative improvement obtained for segment-wise is even significant where our model outperforms FlowNetS by 17.33% obtaining 0.938 EPE. The optical flows estimated by the two models in comparison to the ground truth flows are illustrated by Fig. 3.

A similar result was observed on the Sub UCF I-Frames dataset as the proposed FlowCaps-S model achieved an average testing EPE value of 2.2037, after outperforming the FlowNetS model performance by a notable margin of 4.80%. A visual inspection of generated optical flow shown in Fig. 4 indicates the superiority of our method. Preserving the trend, the FlowCaps-S model outperformed the FlowNetS model by a comprehensive margin of 14.01% while achieving an impressive average testing EPE value of 0.3806. Fig. 5 illustrates the optical flows estimated by the two models on the UTI P-Frames dataset. Hence, it was evident that across all the datasets, the proposed FlowCaps-S model outperformed the FlowNetS model for the optical flow estimation task. Furthermore, it is interesting to note that, the percentage improvement is inversely proportional to both the complexity of the dataset and the number of training samples, suggesting that the proposed CapsNet-S model better generalizes with less number of training data, and on less complex datasets, in comparison to the FlowNetS model. Most interestingly, segment-wise model consistently outperform frame-wise model indicating the advantage of our 3D convolution-based capsule encoder and better exploitation of contextual motion information. Both FlowNetS and FlowCaps-S benefit from segment-wise model, however the improvements obtain by our model is better than the FlowNetS.

As shown in Table 3, for action classification task from the estimated optical flows, the results follow the same pattern as the optical flow estimation task, where the flows estimated with the proposed FlowCaps-S model outperformed those of the FlowNetS model by significant margins. Quantitatively, the proposed FlowCaps-S model contributed to achieving $65\%, 91.20\%$ and 86.02% on the KTH-I Frames, Sub UCF I-Frames and UTI-P Frames datasets respectively, while outperforming its counterpart model by respective margins of $3.70\%, 5.50\%$ and 1.90%. Similar trends can be seen for segment-wise model. Hence, it can be concluded that the optical flows estimated by the proposed FlowCaps-S model can be better-adopted to other tasks such as action classification, than those by the FlowNetS model.

Furthermore, it is interesting to observe that ground truth action recognition performance is sometimes lower than our model. However, we do not expect this behavior on more challenging datasets. Our model is a learning-based optical flow estimation method. Essentially, the model learns about

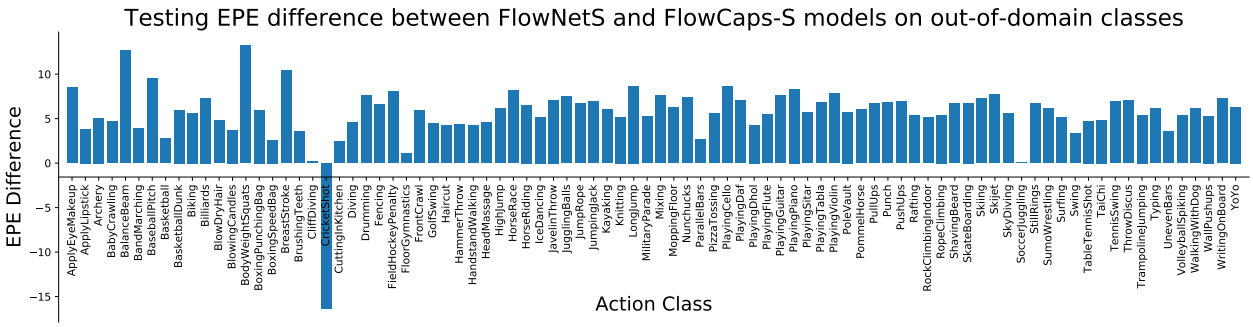
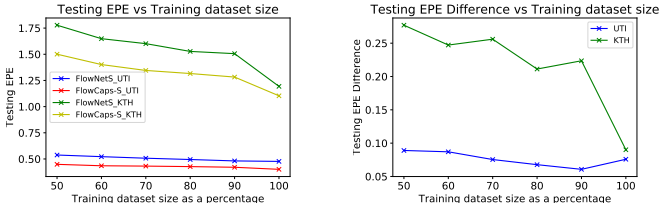


Figure 6. Testing EPE value differences between the FlowNetS and FlowCaps-S model performances on the out-of-domain action classes of the UCF I-frames dataset.



(a) (b)

Figure 7. The comparison of testing EPE values achieved by the FlowNetS and FlowCaps-S models: (a) Testing EPE vs Training dataset size; (b) Testing EPE difference vs Training dataset size.

motion information through entire dataset and hence able to capture dataset specific biases as well. Therefore, we hypothesise that good optical flow learning models might be able to exploit motion information and biases in a dataset and may be able to output flow images that contain more motion information suitable for action recognition. Furthermore, segment-wise approach with a 3-dimensional architecture outperforms the frame-wise approach for both optical flow estimation and action classification tasks across almost all datasets. We conclude that perhaps segment-wise approach is better for optical flow estimation and action recognition.

4.4. Other strengths of FlowCaps

FlowCaps is able to generalize to out-of-domain using fewer training samples. Here, we consider the I-frames extracted from the entire UCF-101 dataset. We test with both models on all the classes of UCF-101 except for classes with no videos containing more than 5 I-frames, and for the five classes considered for training in the Sub UCF I-Frames dataset, which yields 88 out-of-domain action classes.

Fig. 6 illustrates the differences in the testing EPE values obtained from the FlowNetS and FlowCaps-S models. It is evident from observation that except for the *Cricket Shot* action class, the proposed FlowCaps-S model achieves a lower testing EPE than the FlowNetS model in rest of the 87 out-of-domain classes, suggesting that the proposed FlowCaps-S model generalizes to out-of-domain optical flow estimation better than the FlowNetS model.

Our model learns well with a low amount of training data. We hypothesize that when the fraction of the training dataset used decreases, the difference between the average testing EPEs of FlowNetS and FlowCaps-S models should increase, indicating better generalization of the proposed FlowCaps-S model with less training data. To this end, we train the models with fractions of the training data ranging from $50 - 100\%$ with 10% intervals for the KTH-I Frames and UTI P-Frames datasets, and plot the raw EPE values and the corresponding differences in Fig. 7 (a) and Fig. 7 (b) respectively. It is evident from observation that except for the UTI P-Frames dataset instance with the full training set, the rest of the instances favor our hypothesis, across both datasets. Hence, we concluded that the FlowCaps-S model generalizes better than the FlowNetS model for optical flow estimation, with less training data.

5. Conclusion

In this paper we have investigated a deep model for optical flow estimation by extending FlowNetS [6] and Capsule Networks [28] coined FlowCaps-S. We investigated frame-based model and a video segment-based model that utilizes 3D convolutions. We consistently outperform several state-of-the art models for both optical flow estimation on classical benchmarks and on some important action recognition datasets. Interestingly, our model have only a fraction of parameters compared to other baselines. We demonstrate that our model is able to learn with few examples and generalize to out-of-domain examples better than other counterparts.

Acknowledgment

This research/project is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG-RP-2019-010).

References

[1] Christian Bailer, Bertram Taetz, and Didier Stricker. Flow fields: Dense correspondence fields for highly accurate large displacement optical flow estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4023, 2015.

[2] Aviram Bar-Haim and Lior Wolf. Scopeflow: Dynamic scene scoping for optical flow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7998–8007, 2020.

[3] Hakan Bilen, Basura Fernando, Efstratios Gavves, and Andrea Vedaldi. Action recognition with dynamic image networks. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2799–2813, 2017.

[4] Hakan Bilen, Basura Fernando, Efstratios Gavves, Andrea Vedaldi, and Stephen Gould. Dynamic image networks for action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2016.

[5] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *European conference on Computer Vision*, pages 611–625. Springer, 2012.

[6] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2758–2766, 2015.

[7] Kevin Duarte, Yogesh Rawat, and Mubarak Shah. Videocapsulenet: A simplified network for action detection. In *Advances in Neural Information Processing Systems*, pages 7610–7619, 2018.

[8] Basura Fernando, Efstratios Gavves, José Oramas, Amir Ghodrati, and Tinne Tuytelaars. Rank pooling for action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 39(4):773–787, 2016.

[9] Basura Fernando, Efstratios Gavves, Jose M Oramas, Amir Ghodrati, and Tinne Tuytelaars. Modeling video evolution for action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5378–5387, 2015.

[10] Ruohan Gao, Bo Xiong, and Kristen Grauman. Im2flow: Motion hallucination from static images for action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5937–5947, 2018.

[11] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[12] Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 244–253, 2019.

[13] Samitha Herath, Basura Fernando, and Mehrtash Harandi. Using temporal information for recognizing actions from still images. *Pattern Recognition*, 96:106989, 2019.

[14] Geoffrey E Hinton, Alex Krizhevsky, and Sida D Wang. Transforming auto-encoders. In *International Conference on Artificial Neural Networks*, pages 44–51. Springer, 2011.

[15] Berthold KP Horn and Brian G Schunck. Determining optical flow. In *Techniques and Applications of Image Understanding*, volume 281, pages 319–331. International Society for Optics and Photonics, 1981.

[16] Tak-Wai Hui, Xiaoou Tang, and Chen Change Loy. Liteflownet: A lightweight convolutional neural network for optical flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8981–8989, 2018.

[17] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2462–2470, 2017.

[18] Ayush Jaiswal, Wael AbdAlmageed, Yue Wu, and Premkumar Natarajan. Capsulegan: Generative adversarial capsule network. In *Proceedings of the European Conference on Computer Vision*, pages 0–0, 2018.

[19] Vinoj Jayasundara, Sandaru Jayasekara, Hirunima Jayasekara, Jathushan Rajasegaran, Suranga Seneviratne, and Ranga Rodrigo. Textcaps: Handwritten character recognition with very small datasets. In *2019 IEEE Winter Conference on Applications of Computer Vision*, pages 254–262, 2019.

[20] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black. Towards understanding action recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, 2013.

[21] Rodney LaLonde and Ulas Bagci. Capsules for object segmentation. *arXiv preprint arXiv:1804.04241*, 2018.

[22] Pengpeng Liu, Michael Lyu, Irwin King, and Jia Xu. Selflow: Self-supervised learning of optical flow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4571–4580, 2019.

[23] Jathushan Rajasegaran, Vinoj Jayasundara, Sandaru Jayasekara, Hirunima Jayasekara, Suranga Seneviratne, and Ranga Rodrigo. Deepcaps: Going deeper with capsule networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10725–10733, 2019.

[24] Anurag Ranjan and Michael J Black. Optical flow estimation using a spatial pyramid network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4161–4170, 2017.

[25] Jerome Revaud, Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. Epicflow: Edge-preserving interpolation of correspondences for optical flow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2015.

[26] Cristian Rodriguez, Basura Fernando, and Hongdong Li. Action anticipation by predicting future dynamic images. In

Proceedings of the European Conference on Computer Vision, pages 0–0, 2018.

[27] Michael S Ryoo and JK Aggarwal. Ut-interaction dataset, icpr contest on semantic description of human activities (sdha). In *IEEE International Conference on Pattern Recognition Workshops*, volume 2, page 4, 2010.

[28] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems*, pages 3856–3866, 2017.

[29] Christian Schuldt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: a local svm approach. In *Proceedings of the IEEE International Conference on Pattern Recognition*, volume 3, pages 32–36, 2004.

[30] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, pages 568–576, 2014.

[31] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[32] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8934–8943, 2018.

[33] Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui, and Cordelia Schmid. Deepflow: Large displacement optical flow with deep matching. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1385–1392, 2013.

[34] Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1983–1992, 2018.

[35] Yongheng Zhao, Tolga Birdal, Haowen Deng, and Federico Tombari. 3d point capsule networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1009–1018, 2019.

[36] Xizhou Zhu, Yuwen Xiong, Jifeng Dai, Lu Yuan, and Yichen Wei. Deep feature flow for video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2349–2358, 2017.

[37] Shay Zweig and Lior Wolf. Interponet, a brain inspired neural network for optical flow dense interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4563–4572, 2017.