Self-Supervised Approach for Facial Movement Based Optical Flow

Muhannad Alkaddour, Usman Tariq, *Member, IEEE*, and Abhinav Dhall, *Member, IEEE*

Abstract—Computing optical flow is a fundamental problem in computer vision. However, deep learning-based optical flow techniques do not perform well for non-rigid movements such as those found in faces, primarily due to lack of the training data representing the fine facial motion. We hypothesize that learning optical flow on face motion data will improve the quality of predicted flow on faces. This work aims to: (1) exploring self-supervised techniques to generate optical flow ground truth for face images; (2) computing baseline results on the effects of using face data to train Convolutional Neural Networks (CNN) for predicting optical flow; and (3) using the learned optical flow in micro-expression recognition to demonstrate its effectiveness. We generate optical flow ground truth using facial key-points in the BP4D-Spontaneous dataset. This optical flow is used to train the FlowNetS architecture to test its performance on the Extended Cohn-Kanade dataset and a portion of the generated dataset. The performance of FlowNetS trained on face images surpassed that of other optical flow CNN architectures. Our optical flow features are further compared with other methods using the STSTNet micro-expression classifier, and the results indicate that the optical flow obtained using this work has promising applications in facial expression analysis.

Index Terms—Optical flow, deep learning, micro-expression detection, facial expression analysis

1 Introduction

Facial expressions are generated due to non-rigid movement in faces. From the perspective of automatic facial expression recognition (FER), the motion information has been well explored for the task of both micro and macro expression analysis. Optical flow is used to estimate the motion of sets of pixels across images. This information on faces can help characterize both micro and macro expressions, which are useful in expression recognition. A major motivation for using the motion information for FER is based on what is known as the facial feedback hypothesis [1], which, in summary, suggests that facial actions can both encode current emotions as well as *induce* or amplify emotions. An example of this would be that the furrowing of the brow could increase anger [1]. It has also been demonstrated that some facial muscle movements are linked to the compound facial expression of negation [2]. Also, the relation between motion information extracted from the eyes and mouth has been studied in its association with the facial expressions of psychopaths [3]. Facial and head movements are also important in social contexts, such as head motion used to indicate particular social cues, or the famous twitching of the lip corners that may suggest lying [4].

Faces have a peculiar structure. Hence, in this work, we focus on learning optical flow specialized for faces, which we will attempt to constrain the algorithm to learn only lifelike expressions on faces. In doing so, we explore how well a deep network can perform in this task. We demonstrate that the proposed architecture will work well for faces compared and compare it to traditional optical flow algorithms. The results can serve as a precursor to designing motion-based features for supervised and unsupervised learning in tasks such as FER and action unit (AU) recognition. of facial expressions by drawing on existing research linking facial motion information to facial expression and emotion recognition. Several works document the use of facial optical flow features for facial expression recognition and action unit recognition tasks.

We use the BP4D-Spontaneous dataset [5] consisting of videos of 41 participants with different facial expressions to generate the ground-truth optical flow between every pair of consecutive frames in the dataset. The ground-truth optical flow is obtained using facial key-points and image warping with affine transformations. We then use this facial optical flow ground truth to train a convolutional autoencoder based architecture, FlowNetS [6] (specialized for optical flow estimation), to learn optical flow specialized for facial motions, meaning that the motion learned should exhibit local coherency as would be expected on faces. We also modify the architecture by adding a cyclic loss to help the network reconstruct the latter image in a given image pair using the optical flow predicted by the network. We argue that adding this reconstruction in the learning framework improves the predicted optical flow by guiding it using the structure of the image pairs. We perform an ablation study with different loss functions, and compare the

• *Muhannad Alkaddour and Usman Tariq are with the American University of Sharjah, Sharjah 26666, UAE. E-mail: {b00059796, utariq}@aus.edu.*
• *Abhinav Dhall is with the Indian Institute of Technology Ropar, Rupnagar, Punjab 140001, India, and also with the Monash University, Clayton, VIC 3800, Australia. E-mail: abhinav@iitrpr.ac.in.*

Manuscript received 19 March 2021; revised 18 July 2022; accepted 24 July 2022. Date of publication 10 August 2022; date of current version 15 November 2022.
This work was supported in part by the FRG17-R44 research grant, a professional development grant from the College of Engineering, and the Open Access Program under Grant OAPCEN-1410-E00085, all from the American University of Sharjah. This paper represents the opinions of the authors and does not mean to represent the position or opinions of the American University of Sharjah. The work of Dr. Dhall was supported by Australian Research Council under Grant DP190102919.
(Corresponding author: Usman Tariq.)
Recommended for acceptance by A. A. Salah.
Digital Object Identifier no. 10.1109/TAFFC.2022.3197622

performance of our network and other baseline optical flow CNNs by testing on both the Extended Cohn-Kanade dataset [7] and a portion of BP4D-Spontaneous dataset. Finally, we test the usefulness of our network by using the learned optical flow predictions for micro-expression detection using optical flow and the Shallow Triple Stream Three-dimensional CNN (STSTNet) [8].

Hence, the contributions of this paper are:

- Introduction of a*"noisy"* optical flow dataset for faces, making use of the peculiar structure of faces. The *noisiness* of the data comes from the sparse triangulation over the 68 facial landmarks that are used to generate the dataset.
- Learning a network for optical flow estimation, specialized for movements induced by facial expressions. We then complement the structure with a cyclic loss. Our modified architecture outperforms several other networks used for optical flow estimation.
- Exhibiting the usefulness of our trained network by applying it for micro-expression detection.

The remainder of the paper is organized as follows. Section 2 contains related literature in the relevant topics. Section 3 describes the details of the automatic dataset generation used in this paper, while details of the networks trained on the generated dataset are explained in Section 4. The results of the ablation study and micro-expression recognition are presented in Section 5. And finally, we present the concluding remarks and recommendations for improvement and future work in Section 6.

2 Related Work

First, we discuss works related to optical flow estimation using classical and deep learning techniques, along with some of the common challenges. We follow this up by a survey of optical flow methods as applied to faces in particular, and how optical flow is used in tasks such as micro-expression detection.

2.1 Optical Flow Estimation

Optical flow in images is used to estimate the motion of sets of pixels across images. Classical methods, such as [9] and [10], use the intensity derivatives and energy methods to estimate the optical flow.

2.1.1 Optical Flow Challenges

Over the last four decades, notable challenges have been identified in optical flow generation and the methods were specifically developed to overcome them. Shah and Xuezhi [11] provide an extensive review on each challenge, which include motion discontinuities, motion blur and occlusions, brightness, and large motions.

In-the-wild datasets are prone to occlusions in their scenes, since an uncontrolled environment is likely to contain moving objects that overlap in the video sequences. No one method is sufficient to solve the problem in general. Some popular solutions are reviewed in [11], which are image warping and bidirectional inconsistency. Janai et al. [12] approach the problem by considering a triplet instead of a pair of frames and a *photometric* loss to handle the occlusions. Meister et al. [13] build on these concepts by applying their own loss function to improve results of unsupervised learning of optical flow, as well as learning the flow in the forward and reverse directions as in [12], and Ren et al. [14] also use a consistency loss to mitigate the occlusion.

Motion discontinuities can arise in occluded settings and in non-rigid motion, and can result in erroneous optical flow continuation in regions of discontinuity [11]. Sun et al. [15] devise a *non-local term* that assists the objective function in accounting for motion discontinuity. Tian et al. [16] modify this non-local term in a CNN-based method to account for discontinuity in their objective. In addition, Wang et al. [17] adapt it as a CNN block for the same purpose. Other existing approaches include detecting the discontinuous boundary and correcting for the flow [18] and a suitable energy-minimization [19].

2.1.2 Deep Learning for Optical Flow Estimation

With the surge and success of deep learning applications in the past decade, there has also been a rise in using convolutional neural networks to learn optical flow, beginning with the groundbreaking work of Fischer et al. [6] with their *FlowNet* CNN architecture. Building on the success of FlowNet, *FlowNet2.0* [20] was introduced a few years later to improve performance by stacking networks, scheduling the training data, and learning on small-motion datasets. *FlowNet3.0* [21] was also proposed afterwards for scene flow estimation. For our experiments, we use the *FlowNetS* architecture adapted from [6] to train on our dataset. By demonstrating how we can adapt FlowNetS to perform well on datasets consisting of only faces, we can later improve even further by training more advanced architectures on such datasets.

While FlowNet is one of the most popular optical flow deep learning architectures, several other architectures have since been proposed to deal with certain challenges.

Sun et al. [22] used the pyramid-structure CNN architecture *PWC-Net* for optical flow prediction, which we use in this work to test on the face optical flow dataset as a benchmark implementation and compare with our performance. Another optical flow CNN we use for comparison in this work is *LiteFlowNet* by Hui et al. [23], which surpassed Flownet2.0's performance on the KITTI and Sintel final datasets.

In their pioneering work, Zhu et al. [24] developed the *cycleGAN*, which is a type of generative adversarial network (GAN), that implements a cyclic loss function which is used as a metric to evaluate the network's prediction as compared with one of the inputs. Both Yu et al. [25] and Lai et al. [26] adapt the cyclic loss for optical flow learning. Both of the latter architectures used a differentiable spatial transformer layer with learnable parameters, adapted from Jaderberg et al. [27].

2.2 Optical Flow and Facial Expression Analysis

We now discuss various optical flow methods as applied to facial expression analysis, many of which are based on deep networks. One important work in learning optical flow for facial expressions, by Snape et al. is *Face Flow* [28], which minimizes a proposed energy to learn the flow field for a sequence of frames consisting of facial expressions. Another relevant work is optical flow dataset generation done by Le et al. [29] who are also concerned with producing optical flow ground-truth data for general video sequences. According to them, little prior work exists on how the performance of CNNs is influenced by optical flow datasets, and their main focus is that of non-rigid motion. Our work can be considered to be a contribution to the study of optical flow's effects on CNNs, with the difference being that we focus on facial datasets instead. We attempt to learn optical flow from the face movements themselves. On a side note, an evaluation of different optical flow techniques applied for facial expression recognition can be found in [30].

We mention a few implementations of deep networks in facial expression analysis using optical flow. Koujan et al. [31] recently proposed *DeepFaceFlow*, in which they construct a 3D optical flow dataset for faces from a large collection of videos and compare the performance of their U-net trained on their dataset with other CNN architectures for both 2D and 3D optical flow estimation. One key difference between our work and theirs is that we incorporate a cyclic loss to test how well the flow fields reconstruct the second image in the image pairs. Additionally, the training data we generate is based upon the BP4D-Spontaneous dataset, which is specifically tuned to exhibit various emotions and thus more specialized for expression recognition tasks. In addition, we also test our network's performance on micro-expression detection.

Several works also use optical flow for action unit recognition. Ma et al. [32] proposed Action Unit (AU) R-CNN to improve AU recognition by using expert prior knowledge, which can be in the form of optical flow, to guide an R-CNN in locating the action region. Yang and Yin [33] learn both optical flow and facial action units for static images in one combined CNN architecture. They learn optical flow in an unsupervised manner, as an intermediate output of a deep model (*OFNet*), which when combined with the first image in a pair (also the input of the model), gives the second reconstructed image at the final output. They call this intermediate output as coming from a *hidden layer*. They use the output of this hidden layer as an input to another network (*AU-Net*) to detect facial AUs. They train both AU-Net and OFNet jointly.

In another work that uses a cyclic loss, Li et al. [34] learn a symmetric encoder-decoder architecture to learn AU representations in a semi-supervised manner. They train it with pairs of face images of the same person in a video with different facial actions and head poses. Hence, these images are not coming from consecutive frames. They attempt to disentangle the embeddings related to head pose and action units, by constraining the pixel movements related to the AUs to be subtle, compared to those related to head-pose. They then use the learned AU and pose related displacements to reconstruct the second image in an image pair, given the first one. After learning, they use the AU embeddings for facial AU detection. Note, that since embeddings are being learnt here on frames of a person that are not consecutive, within a video, these embeddings will not be able to learn the subtle pixel movements that happen within a certain AU. Other works that use optical flow for action unit recognition can be found in [35], [36], and [37]. Our work, on the other hand, focuses on learning optical flow specialized for faces. We introduce a noisy optical flow dataset, that we generate using the motion of sparse facial landmarks. We then learn a network for optical flow estimation, specialized for movements induced by facial expressions. We then complement the structure with a cyclic loss. We show that our modified architecture outperforms several other networks used for optical flow estimation. In addition, we demonstrate the usefulness of our approach by applying it for micro-expression detection.

Liong et al. [38] exploit the optical flow in a video sequence between the frame with the highest intensity, called apex, and each of the rest of the frames, using the optical flow as input to a deep network for micro-expression detection. They also use apex and onset frames in [8] to compute optical flow along with an added feature, the optical strain, as input to STSTNet, which we adapt in this work to test for micro-expression recognition. Verburg and Menkovski [39] use optical flow histograms as feature inputs to a recurrent neural network for the recognition of micro-expressions. Li et al. [40] use a CNN to locate facial keypoints and FlowNet2.0 to compute optical flow, and the flow features are then used with a support vector machine for micro-expression detection.

After having reviewed several related works, we now describe the dataset preparation in our work.

3 Dataset Preparation

Our method is inspired by the progress in self supervised learning techniques for action recognition [41] and eye gaze prediction [42]. We use the BP4D-Spontaneous dataset [5], which consists of 41 subjects with 8 video sequences each, containing videos of elicited emotions. The motivation for using BP4D-Spontaneous is its inclusion of both head and facial motion. While local non-rigid facial motion estimation is the primary focus of this work, it is also useful to capture this local facial flow in the presence of head motion. Since BP4D-Spontaneous is concerned with spontaneously elicited expression sequences and 3D encoding, more general motion is available. Other datasets, such as the Extended CK+ [7], are more specialized for AU or facial expression detection, and thereby are less suited for a more general motion framework. Moreover, this allows us to test how optical flow performs on micro-expression detection when trained on a dataset not specialized for micro-expression detection.

Fig. 1 shows the overall pipeline for a pair of frames and how they can be used for dataset generation and CNN training.[1]

We introduce the notation that we'll use throughout this section to generate the optical flow ground truth from the BP4D-Spontaneous dataset [5]. For a given sequence S in the dataset, we denote the frames contained in S by $F = \{f_k\}_{k=0}^{N_f}$, where $f_k \in \mathbb{R}^{H\times W\times 3}$ are the ordered frames. Our aim in this section is to compute a set of optical flow fields, $\boldsymbol{U}$ separately for every ordered set of frames, F, where $\boldsymbol{U} = \{\boldsymbol{u}_k\}_{k=0}^{N_f-1}$ contains the optical flow fields $\boldsymbol{u}_k : \mathbb{R}^{H\times W} \mapsto \mathbb{R}^{H\times W\times 2}$ for each frame except the final one in that sequence. The $\boldsymbol{u}_k$ are vector-valued functions defined on the image grid.

Landmarks $\boldsymbol{P}$ on the face in S are tracked for each frame using the open source OpenFace pipeline [43], which uses

1. Our code implementing the algorithm in this section will be made publicly available at https://github.com/malkaddour/Self-Supervised-Approach-for-facial-movement-based-optical-flow

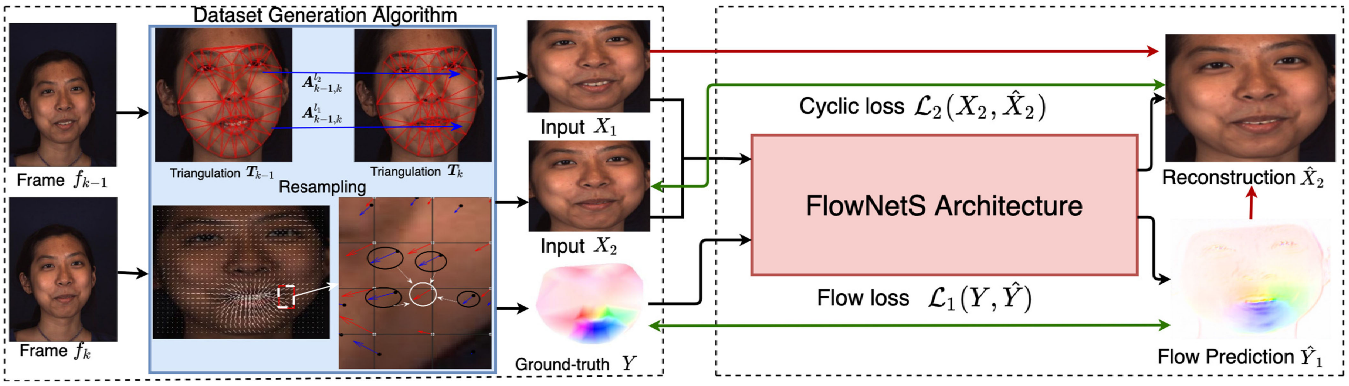

Fig. 1. Overall pipeline for data generation and network training: Two examples of the affine maps are shown for some triangles l_1, l_2, and an illustration of the resampling process is shown on a 3×3 grid of a portion of the optical flow field.

the *Convolutional Experts Constrained Local Model* [44] to obtain 68 landmarks perfor the kth face, $\boldsymbol{P}_k = (\boldsymbol{p_0} \boldsymbol{p_{68}})^T_k \in \mathbb{R}^{68\times2}$, where T denotes the transpose operation.

Next, we use Scipy's Delaunay triangulation package to obtain a triangular mesh over $\boldsymbol{P}_0$ in the first face f_0. This mesh divides the face in f_0 into N_t disjoint triangles $\boldsymbol{T}_0 = \{\boldsymbol{t}^l\}_{l=0}^{N_t}$, where each $\boldsymbol{t}^l = (\boldsymbol{v}_0, \quad \boldsymbol{v}_1, \quad \boldsymbol{v}_2)^T_l \in \mathbb{R}^{3\times2}$ is the matrix with rows composed of vertices of triangle l. After triangulating f_0, we use similar triangulation on the remaining frames in the sequence, yielding the set of triangulations $\{\boldsymbol{T}_k\}_{k=0}^{N_f}$ on S.

We use the triangulation $\boldsymbol{T}_{k-1}$ to capture the local motion on every triangle in the face partition from frame f_{k-1} to frame f_k. Given the triangle $\boldsymbol{t}^l_{k-1}$, we infer an affine map $\boldsymbol{A}^l_{k-1} \in \mathbb{R}^{3\times3}$ that sends its vertices to the vertices in $\boldsymbol{t}^l_k$. Specifying three mappings are sufficient to uniquely define an affine map [45]. We can define $\boldsymbol{t}^{l,*} = (\boldsymbol{t}^l, \quad \mathbf{1}_{3\times1})^T \in \mathbb{R}^{3\times3}$ to be the matrix of homogeneous coordinates of each vertex. Then, for all triangles in f_{k-1} and f_k, $\boldsymbol{A}^l_{k-1}$ that sends $\boldsymbol{t}^{l,*}_{k-1}$ to $\boldsymbol{t}^{l,*}_k$ is uniquely determined by,

$$\boldsymbol{A}^{\boldsymbol{l}}_{\boldsymbol{k-1}} = \left(\boldsymbol{t}^{l,*}_{k-1}\right)^{-1} \boldsymbol{t}^{l,*}_k. \tag{1}$$

This gives the required matrix for the affine map. Note that if the triangle is degenerate, then $\boldsymbol{t}^*_{k-1}$ will be singular. Once the correspondence between the two triangles across frames is known, $\boldsymbol{A}^l_{k-1}$ also maps the interior of $\boldsymbol{t}^l_{k-1}$ to the interior of $\boldsymbol{t}^l_k$, since barycentric coordinates [46] are invariant under affine maps [45].

We use the barycentric coordinates to compute the interiors of all the triangles in $\boldsymbol{T}_0$, and then learn each affine map $\boldsymbol{A}^l_0$ as described above to map all the triangle interiors from $\boldsymbol{T}_0$ to $\boldsymbol{T}_1$. To compute the interior of the triangle using barycentric coordinates, an efficient algorithm from [47] can be used to test if an arbitrary point $\boldsymbol{v}$ is contained in a given triangle by solving,

$$\boldsymbol{V}\boldsymbol{\lambda} = \boldsymbol{b}, \text{ where,}$$
$$\boldsymbol{V} = \begin{pmatrix} \|\boldsymbol{v}_1 - \boldsymbol{v}_0\|_2^2 & (\boldsymbol{v}_2 - \boldsymbol{v}_0)\cdot(\boldsymbol{v}_1 - \boldsymbol{v}_0) \\ (\boldsymbol{v}_2 - \boldsymbol{v}_0)\cdot(\boldsymbol{v}_1 - \boldsymbol{v}_0) & \|\boldsymbol{v}_2 - \boldsymbol{v}_0\|_2^2 \end{pmatrix}, \text{ and}$$
$$\boldsymbol{b} = \begin{pmatrix} (\boldsymbol{v} - \boldsymbol{v}_0)\cdot(\boldsymbol{v}_1 - \boldsymbol{v}_0) \\ (\boldsymbol{v} - \boldsymbol{v}_0)\cdot(\boldsymbol{v}_2 - \boldsymbol{v}_0) \end{pmatrix} \tag{2}$$

for $\boldsymbol{\lambda} = (\lambda_1, \quad \lambda_2)^T$, and $\lambda_3 = 1 - \lambda_1 - \lambda_2$ [47]. and if the If each $\lambda_i \in [0, 1]$, then $\boldsymbol{v}$ lies in the closure of the triangle of interest, i.e., $\boldsymbol{v}$ is a convex combination of the columns of $\boldsymbol{t}$. We test all points in this way using a rectangular discrete grid surrounding the triangle. Repeating this for all f_k in the sequence is overall computationally expensive, so we only do it for triangles in the first frame of that video. By invariance of barycentric coordinates under the affine maps $\boldsymbol{A}^l_{k-1}$, this also determines the barycentric coordinates for all subsequent frames f_k, $k > 0$.

After determining the affine maps and mapping the triangles and their interior pixels $\boldsymbol{v}_{k-1}$ to $\boldsymbol{v}_k$, we compute the per-pixel optical flow vector $\tilde{\boldsymbol{u}}_{k-1}$ by

$$\tilde{\boldsymbol{u}}_{k-1} = \boldsymbol{v}_k - \boldsymbol{v}_{k-1}. \tag{3}$$

However, when the domain is not a discrete grid, the optical flow fields $\tilde{\boldsymbol{u}}_k$ are defined on points that are not necessarily pixel coordinates, which affects the frames *after* f_0. The optical flow field $\tilde{\boldsymbol{u}}_{k-1}$ from Equation (3) is defined on a discrete grid, but the pixels that are mapped from f_0 to f_1 will subsequently be mapped from f_1 to f_2, in which case it is not guaranteed that they also lie on a discrete grid. To recover the optical flow field $\boldsymbol{u}_{k-1}$ on a discrete grid in the target image, we use bicubic spline interpolation over the irregular grid using $\tilde{\boldsymbol{u}}_{k-1}$. We only do this to define the optical flow field at each frame, but continue to learn the affine maps on the irregular grids, since we wish to preserve the same barycentric coordinates obtained in f_0 for all frames. The flow fields are stored in .flo formats for later use in the experiments.

Together with the resampling stage, this procedure gives us the ground-truth vector field for all pixels of frame f_{k-1}. The details can be summarized as follows:

1) Starting from frame f_0, determine the interiors of all triangles $\boldsymbol{t}^l_0$, using barycentric coordinates.
2) Learn the affine maps sending all $\boldsymbol{t}^l_0$ to $\boldsymbol{t}^l_1$ and transform the entire face to obtain the first optical flow field $\boldsymbol{u}_0$.
3) For all frames starting from f_1, again infer the affine maps sending all $\boldsymbol{t}^l_1$ to $\boldsymbol{t}^l_2$ and apply the transformation on all the pixels which have already been mapped from frame f_0. This removes the need to expensively compute the triangle interiors for frame f_1 while still finding the optical flow field $\tilde{\boldsymbol{u}}_1$.
4) From $\tilde{\boldsymbol{u}}_1$, resample the flow field over a discrete grid to yield the ground-truth flow $\boldsymbol{u}_1$.

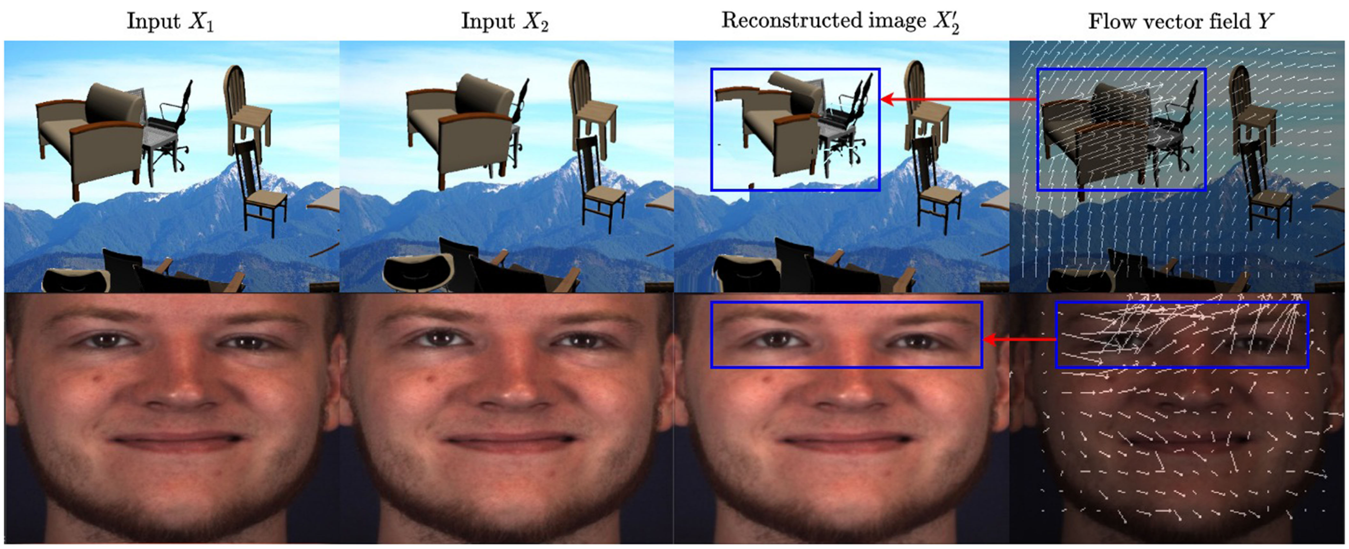

Fig. 2. Effect of using flow field Y to warp X_1 to X_2' is demonstrated for images with large (top) and small (bottom) motion. Flow vector magnitudes are not to scale and are amplified for illustrative purposes.

5) Repeat steps 3-4 for the remaining frames in a sequence $\{f_k\}_{k=2}^{N_f}$, for all sequences and subjects.

The total number of images in the generated dataset is 325720, and these were partitioned into 228171, 65130, and 32419 for training, validation, and test data respectively. The dataset generation was completed in a total of about five days using multicore CPU parallel processing with four parallel processes running at a time. Our method generates optical flow that captures the motion induced by facial expressions but is noisy due to the relatively sparse number of landmarks used to triangulate the face. However, the overall mapping from one frame to the next is a piecewise-affine map that can capture the discontinuous nature of facial expression motion, since there are no continuity restrictions imposed between a given mesh and its neighbors. Of course, smaller and denser meshes, such as those around the eyes and mouth in our triangulation, can better detect any nonlinear or discontinuous motion that may arise. An improvement to our method would be to increase the sampling of landmarks and allow for finer resolution in the piecewise-affine map.

4 Baseline Networks

In this section, we describe the CNN architecture used to train the optical flow, followed by the training and ablation study details. These details include the different hyperparameters used in the different experimental setups, such as the choices of loss functions, the loss weights, and training/testing data split.

4.1 CNN Architecture: *FlowNetS*

To test the effects of having a large, *"noisy"* ground-truth optical flow dataset specialized for faces on CNNs, the FlowNetS [6] architecture was used. FlowNetS is one of the pioneering CNNs on optical flow learning. While more sophisticated optical flow architectures have been developed, our purpose is to demonstrate the improvement of training a CNN with face data compared with some other datasets, e.g., the FlyingChairs dataset, as a proof of concept. Should we discover an improvement, we can expand it in the future to tackle other problems (e.g., robustness to occlusions).

FlowNetS is a convolutional autoencoder architecture which accepts a pair of images as input and outputs the per-pixel optical flow from the first image to the second. It consists of a sequence of downsampling convolutional layers in the encoder followed by upsampling layers in the decoder, in addition to intermediate operations and concatenations. Another variant of FlowNet, which is FlowNetCorr, is characterized by a cross-correlation layer which fuses two input streams together, contrasted to FlowNetS which combines them with a simple concatenation. The difference in performance reported in [6] is not too significant, and including the cross-correlation layer during training resulted in the inconvenience of much longer training times.

The output resolutions of each of the flow predictions in our network are slightly different than the original FlowNetS. Specifically, the ratio of our flow prediction heights to theirs is 24:17, and our widths to theirs is 4:5. The reader is referred to [6] for specific details on the network architecture.

4.2 Cyclic Loss for Image Reconstruction

For some of the experiments described in the next section, a cyclic loss is implemented to minimize the difference between the output predicted using the flow prediction and the second input image. This resulted in an additional warping layer to the network that acts on the flow prediction with highest resolution. The warping layer uses the predicted per-pixel flow field vectors to warp the first input image, and the result is recovered using bilinear interpolation. We note that structures inherent only to the second input cannot be reproduced in the warped output, since the warping function only changes pixel locations from the first input, and does not contain any learnable parameters. Fig. 2 shows two examples of this phenomena from FlyingChairs and our face dataset, showing the original input image pair (X_1, X_2), the image X_2' deformed using the flow field, and visualization of the flow field Y.

The dominant motion in the FlyingChairs image pair from the flow field is rightward motion of the left armchair.

The location of the armchair in the warped image is correct, but the reconstruction of the warped portion is missing. This is also present in the smaller desk chair, making a copy of itself at the warped location during reconstruction. Due to these large differences in the images, adding a warping layer while training on the FlyingChairs dataset is likely to worsen the network's performance. However, this effect is much more subtle in our face dataset due to the higher frame rate of the sequences, which causes lower magnitude motion between every two consecutive frames. For the face example in Fig. 2, the deformed image X_2' is perceptually similar to the actual X_2, particularly in the upwards motion of the eyes and the slight rightward motion caused by the furrowing of the brow. Since the time difference between two frames is very small in the face dataset, it is very unlikely for new structures to be introduced in X_2. A notable exception to this is the opening (closing) of the mouth due to revealing (hiding) teeth, which cannot be reproduced by pixel rearrangement alone. Another exception would be the squinting or widening of the eyes for the same reason, since the eyelid or eyeball would not be present in the first image. Although the artifacts caused by the warping produced a flawed image in the FlyingChairs dataset, we hypothesize and show that it still helps guide the directions of the predicted flow when training on faces since the undesirable effects are considerably less due to the lower amount of new structure.

4.3 Training and Ablation Studies Details

The training details of the aforementioned architecture are described in this section[2]. Ablation studies are performed on FlowNetS by training the network with different loss functions and their corresponding weights.

We denote by (X_i, X_{i+1}) the pair of successive input frames, where $X_i, X_{i+1} \in \mathbb{R}^{384\times512\times3}$ and $\boldsymbol{Y}_i = \{(Y_i)_k\}_{k=1}^5$, $\hat{\boldsymbol{Y}}_i = \{(\hat{Y}_i)_k\}_{k=1}^5$ contain the intermediate multi-scale ground-truth and prediction flow fields respectively, where the elements $(Y_i)_k$, $(\hat{Y}_i)_k \in \mathbb{R}^{H_k\times W_k\times 2}$. The i enumerates the entire training set, and successive image frames are input to the network at every iteration. The resolutions of the intermediate flow fields are $(H_k, W_k) = (384 \times 2^{-k}, 512 \times 2^{-k})$ for $k \in \{1, \dots, 5\}$ in the decoder. $(Y_i)_1$, $(\hat{Y}_i)_1$ are the largest flow fields, as in the original FlowNetS network. Note that, in the following, we drop the added subscript and refer to them as Y_i and $\hat{Y}_i$.

Since we assume that the background is stationary, much of the ground-truth flow field outside of the boundaries defined by the key-points are zero vectors. To make the training more practical, we zoom on the box with vertices defined by the key-points with maximal and minimal coordinates plus somean offset of 10 pixels each in the x and y directions. The cropped images and flow fields are then resized using bilinear interpolation. To preserve the units of the flow vectors as pixels, they are scaled accordingly in the horizontal and vertical directions.

Next, we describe the different experimental setups used to train the networks.

2. Source code for training and evaluation, as well as the trained models, will be available at https://github.com/malkaddour/Self-Supervised-Approach-for-facial-movement-based-optical-flow

TABLE 1
The Average EPE for Each Network Described in Section 4.3.1, Trained and Tested on All Three Datasets

	Tested on		
Trained on	Faces	FlyingChairs	Sintel
Faces	0.4054	5.8495	5.1731
FlyingChairs	1.4040	1.4413	3.0300
Sintel	0.8282	7.7613	6.23580

4.3.1 Experimental Setup 1: No Cyclic Loss

In this experiment, the architecture is used without the additional warping layer. The network was trained for 30, 40, and 400 epochs on the face, FlyingChairs, and Sintel datasets respectively, with 15000, 21592, and 870 training and 1000, 640, and 271 validation input image pairs each. The batch size used for training is 16 input pairs. The loss function is the average endpoint error (EPE), $\mathcal{L}_1^i(\boldsymbol{Y}_i, \hat{\boldsymbol{Y}}_i)$, defined for one output by,

$$\mathcal{L}_1^i(\boldsymbol{Y}_i, \hat{\boldsymbol{Y}}_i) = \sum_{k=1}^{5} \frac{w_k}{H_k W_k} \sum_{j=1}^{H_k W_k} \left\| (\boldsymbol{y}_{ij})_k - (\hat{\boldsymbol{y}}_{ij})_k \right\|_2. \tag{4}$$

Here, the w_k are loss weights for each intermediate flow prediction loss, given by $w_k = 2^{-k}$. H_k, W_k are the sizes of the intermediate predictions and the $(\boldsymbol{y}_{ij})_k, (\hat{\boldsymbol{y}}_{ij})_k$ are the flow vectors for the jth pixel of the kth ground-truth and predicted flow fields $(Y_i)_k$ and $(\hat{Y}_i)_k$. The optimizer used is Adam, with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ as in [6]. This performs better than alternative optimizers. We initialized the learning rate α at $1e{-}4$ for faces and $5e{-}5$ for FlyingChairs and scheduled similar to [6].

For preliminary experimentation, we trained the network once on the face data for a 15k and 1k training and validation split. We then used a disjoint set of 3k image pairs from the face dataset as the test set. Also, we used the aforementioned validation sets for FlyingChair and Sintel validation sets as test sets, since we do not have access to their original test sets.

We then tested the model, learnt on 15k training image pairs from the face data, separately on the face, FlyingChairs, and Sintel test sets [48]. Then we repeated this by training on FlyingChairs and Sintel training datasets individually and testing them on all three test sets. We report these results in Table 1.

After the preliminary experiment, we trained the same network again from scratch on faces only for a 228k, 65k, and 32.5k train/val/test split, exactly as in the next two experiments, to make them comparable. The latter setup is referred to as *Experiment 1*, from here onwards.

4.3.2 Experimental Setup 2: With Cyclic Loss

When the warping layer [26] at the end of the network is included, it is necessary to define a cyclic loss function for the warped output $\hat{X}_{i+1}$ and the second input X_{i+1}. We expect to see an improvement in the flow prediction due to the cyclic loss. For this experiment, we define the additional cyclic loss function $\mathcal{L}_2^i(X_{i+1}, \hat{X}_{i+1})$ for one output pair i as:

$$\mathcal{L}_2^i(X_{i+1},\hat{X}_{i+1}) = \frac{1}{HW}\sum_{j=1}^{HW}\frac{1}{3}\sum_{k=1}^{3}\left\|x_{i+1,j,k}-\hat{x}_{i+1,j,k}\right\|_{H_1}$$
$$\|x\|_{H_1} = \begin{cases}\frac{1}{2}x^2 & |x|\le d\\ \frac{1}{2}d^2+d(|x|-d) & |x|>d\end{cases} \quad (5)$$

which uses the Huber loss function $\|x\|_{H_1}$[49], a variant of the L_1 loss that is everywhere differentiable, since it is quadratic for small values of x. The $x_{i+1,j,k}$, $\hat{x}_{i+1,j,k}$ are values of the jth pixel of X_{i+1}, $\hat{X}_{i+1}$ at color channel k. We also note that $\hat{X}_{i+1}$ is a function of the first image of the input pair, X_i, and $\hat{Y}_i$, which is the flow prediction with largest resolution. The total loss function $J(X,\hat{X},Y,\hat{Y})$ for all training pairs is then

$$J(X,\hat{X},\boldsymbol{Y},\hat{\boldsymbol{Y}}) = \frac{1}{M}\sum_{i=0}^{M-1}\left[\lambda_1\mathcal{L}_1^i(\boldsymbol{Y}_i,\hat{\boldsymbol{Y}}_i)+\lambda_2\mathcal{L}_2^i(X_{i+1},\hat{X}_{i+1})\right] \quad (6)$$

with the $\mathcal{L}_1^i, \mathcal{L}_2^i$ defined in Equations (4) and (6) and λ_1, λ_2 to be specified, averaged over all M training examples. In this experiment, we train the network on both faces and FlyingChairs datasets using two different sets of loss weights $\lambda_1,\ \lambda_2$. One network has more emphasis on reconstruction, with $\lambda_2 = 0.6,\ \lambda_1 = 0.4$. We refer to this as Case I. The other network has higher weight assigned to the EPE with $\lambda_1 = 0.75,\ \lambda_2 = 0.25$. We refer to this as Case II. Note that the w_i in Equation (4) should sum to λ_1. For both cases, we trained the network on faces for 15 epochs and 228160 training pairs. Learning rates were kept constant for these experiments throughout training, since scheduling them as previously done lead to very large gradients halfway through training. In Case I, the learning rates were $2.5e{-}6$ and $1.25e{-}6$ for faces and FlyingChairs respectively, and in Case II, they were both set to $2.5e{-}6$. We then tested the trained networks on the test set of 32416 image pairs.

4.3.3 *Experimental Setup 3: With Cyclic Loss, Smoothness Constraint, and Average Angular Error*

In this experiment, we added an additional loss function $\mathcal{L}_3^i(\hat{Y}_i)$. In Case I of this experiment, a smoothness constraint was imposed on the flow prediction by minimizing the flow gradients, defined as:

$$\mathcal{L}_3^i(\hat{Y}_i) = \frac{1}{HW}\sum_{j=1}^{HW}\left(\left\|\frac{\partial\hat{u}_{ij}}{\partial x}\right\|_{H_1}+\left\|\frac{\partial\hat{u}_{ij}}{\partial y}\right\|_{H_1} +\left\|\frac{\partial\hat{v}_{ij}}{\partial x}\right\|_{H_1}+\left\|\frac{\partial\hat{v}_{ij}}{\partial y}\right\|_{H_1}\right) \quad (7)$$

where $(\hat{u}_{ij},\hat{v}_{ij})$ are the components of the predicted flow vector $\hat{\boldsymbol{y}}_{ij}$ at every pixel j.

Another common metric to quantify performance of optical flow algorithms [50] is the average angular error (AAE). The average angular error between two flow vectors is the average of the angle difference between every ground-truth and predicted flow vectors in the homogeneous coordinates, which are $\boldsymbol{y}_j^* = (\boldsymbol{y}_j^T,1)^T$ and $\hat{\boldsymbol{y}}_j^* = (\hat{\boldsymbol{y}}_j^T,1)^T$ respectively. In Case II, the loss function $\mathcal{L}_3^i(Y_i,\hat{Y}_i)$ is defined as:

$$\mathcal{L}_3^i(Y_i,\hat{Y}_i) = \frac{1}{HW}\sum_{j=1}^{HW}\arctan\left(\frac{\left\|\boldsymbol{y}_{ij}^*\times\hat{\boldsymbol{y}}_{ij}^*\right\|_2}{\boldsymbol{y}_{ij}^*\cdot\hat{\boldsymbol{y}}_{ij}^*}\right) \quad (8)$$

The total loss function is then a weighted sum of the loss functions

$$J(X,\hat{X},\boldsymbol{Y},\hat{\boldsymbol{Y}}) = \frac{1}{M}\sum_{i=0}^{M-1}[\lambda_1\mathcal{L}_1^i(\boldsymbol{Y}_i,\hat{\boldsymbol{Y}}_i)+\lambda_2\mathcal{L}_2^i(X_{i+1},\hat{X}_{i+1}) +\lambda_3\mathcal{L}_3^i(Y_i,\hat{Y}_i)] \quad (9)$$

We trained the network on only the faces dataset for 14 epochs and 228160 training pairs, with $\lambda_1 = 0.3$, $\lambda_2 = 0.5$, $\lambda_3 = 0.2$, and learning rate $2.5e{-}6$. We initialized the weights from the results of Experiment 2 (Case I), to see if there is any improvement in flow prediction after adding $\mathcal{L}_3$. In the next sections, we will use abbreviations for experiment and case numbers in the discussions for brevity. For example, Experiment 2, Case II is referred to as Exp. 2II, and no Roman numerals mean we refer to both cases of that particular experiment.

4.4 Micro-Expression Detection

In this section, we describe how optical flow features are used for a micro-expression recognition task to demonstrate the efficacy of the optical flow generated using our method. The use of optical flow in micro-expression recognition has proven useful in several prior works, as described in Section 2.2.

4.4.1 *CNN and Optical Flow Features*

To train the optical flow features, we use the three-dimensional lightweight CNN proposed by Liong et al. [8], named the "Shallow Triple Stream Three-dimensional CNN", or STSTNet, which shows improved results compared with their previous work and other deep networks for micro-expression recognition. Their algorithm is evaluated on the CASME II [51], SAMM [52], and SMIC [53] datasets, composed of videos containing micro-expressions that represent either negative, positive, or surprise emotions (three-class classification). For comparison, we do the same using our optical flow features on the SAMM and SMIC datasets. For each video sequence, they compute the optical flow between the onset and apex frames, and use this optical flow as input to train STSTNet classifier. The apex frames in SAMM are provided with the dataset, which is not the case with SMIC. The apex frames were also used for micro-expression recognition on SMIC dataset by Quang et al. [54]. We make use of their labeling for the SMIC dataset. We crop the faces based on keypoints obtained using the OpenFace 2.0 toolbox [43] for SAMM. For SMIC, since OpenFace failed to detect the keypoints for some images, we instead use the dlib facial landmark detector [55], which is based on an ensemble of regression trees [56], and define the crop border at 15 pixels away from the maximum and minimum x and y image coordinates.

We follow their the recommended approach in [8] to train the STSTNet. The optical flow from the onset to the apex frame is used to compute the optical flow strain $\epsilon(\boldsymbol{U})$ for a given flow field, $\boldsymbol{U} = (u(x,y), v(x,y))$. The strain is

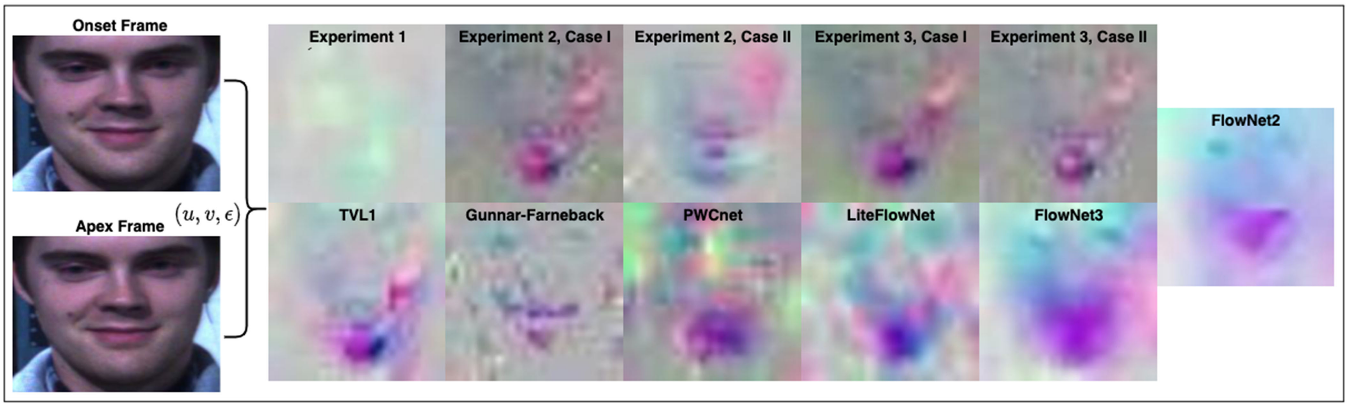

Fig. 3. An example of the computed optical flow features used as inputs to train STSTNet for each network variant. Source: subject 03, SMIC [53].

defined [8] by the symmetric matrix known as the strain tensor

$$\epsilon = \frac{1}{2}\left[\nabla U + (\nabla U)^T\right] = \begin{pmatrix} \frac{\partial u}{\partial x} & \frac{1}{2}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \\ \frac{1}{2}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) & \frac{\partial v}{\partial y} \end{pmatrix}. \quad (10)$$

The strain of a planar displacement field (u, v) is well-known in solid mechanics, consisting of normal strains $\epsilon_{xx}, \epsilon_{yy}$, which are the diagonal elements, and shear strains $\epsilon_{xy} = \epsilon_{yx}$, which are the off-diagonal elements [8]. The strain values represent the type of local deformation that occurs at each point in the flow field. The optical strain norm $||\epsilon(u(x,y), v(x,y))||_s$ is then defined [8] as:

$$||\epsilon(u(x,y), v(x,y))||_s = \sqrt{\epsilon_{xx}^2 + \epsilon_{yy}^2 + 2\epsilon_{xy}^2}, \quad (11)$$

The optical strain feature $V \in \mathbb{R}^{H\times W\times 3}$ is an RGB image and for a given pixel coordinate (x_h, y_h), takes the value $(u(x_h, y_h), v(x_h, y_h), ||\epsilon(u,v)||_s) \in \mathbb{R}^3$. Fig. 3 shows an example of the optical flow feature computed for an image pair, using the optical flow obtained from each network. In this example, the salient motion is an upwards curling of the lips plus a subtle leftwards shift in glance.

4.4.2 Micro-Expression Detection Experimental Setup

The authors of STSTNet [8] evaluate their model using leave-one-subject-out cross-validation (LOSOCV), and we do the same to train the micro-expression recognition networks. The SAMM (normal) and SMIC HS datasets were both used for the task. All optical flow networks described in Section 4.3.2 are used separately to train STSTNet. For every optical flow network, we train the network three times: once on SAMM, once on SMIC, and once on the combined dataset consisting of both. Prelabeled onset and apex frames for each sequence were used to compute the optical flow. The total number of samples across all subjects is 290, which was split into a distinct training and test set for each subject due to the LOSOCV. This caused the number of samples in the test set to vary between the minimum at 1 (0.345%) and the maximum at 37 (12.76%), since the number of samples across subjects was not uniform. We use the publicly available code provided by the authors [8], and thus replicate the exact same network architecture, with a learning rate of 5e−5 and maximum epochs set to 500. We note that the RGB input images, described in Section 4.4, are resized to a resolution of $28 \times 28 \times 3$. We also compute the TVL1 optical flow on SAMM and SMIC, as done in [8], to compare its performance with the optical flow features obtained from other networks.

To deal with the class imbalance, we use macro-averaged recall, precision, and F_1-scores to evaluate the performance of every trained network. Additionally, the metrics specified by Yap et al. [57] are the micro-averaged F_1-score and Unweighted Average Recall (UAR). The definition of UAR is equivalent to macro-averaged recall R_M. UAR is also popular for imbalanced multiclass problems [8]. The performance measures are defined as [58]:

$$R_M = \frac{1}{n}\sum_{i=1}^{n}\frac{\sum_{j=1}^{m} tp_i^j}{\sum_{j=1}^{m} tp_i^j + fn_i^j}, \; R_\mu = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} tp_i^j}{\sum_{i=1}^{n}\sum_{j=1}^{m} tp_i^j + fn_i^j},$$
$$P_M = \frac{1}{n}\sum_{i=1}^{n}\frac{\sum_{j=1}^{m} tp_i^j}{\sum_{j=1}^{m} tp_i^j + fp_i^j}, \; P_\mu = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} tp_i^j}{\sum_{i=1}^{n}\sum_{j=1}^{m} tp_i^j + fp_i^j},$$
$$F_{1_x} = \frac{2P_xR_x}{P_x + R_x}, \quad G_M = \sqrt{[n]}\prod_{i=1}^{n}\frac{\sum_{j=1}^{m} t_{p_i}^j}{\sum_{j=1}^{m} t_{p_i}^j + f_{n_i}^j}. \quad (12)$$

The subscripts M, μ denotes macro and micro-averaging, respectively, and for the F_1-score, $x \in \{M, \mu\}$. tp_i^j, fp_i^j, and fn_i^j denote the true positive, false positive, and false negative of class i, sample j, for a total of $n\,(=3)$ classes and m samples. Note that when the prediction for a given class is a true positive, this also counts as a true negative for each of the other two classes. The macro-averaged metrics tend to remove the bias caused by the imbalance degree, since it does not ”favor” the classes with higher number of examples, as opposed to micro-averaging [58].

Moreover, since LOSOCV is used, this yields one metric per subject. We will combine the metrics to a single scalar, which we will refer to as the *aggregated metric*. This has been done in other works such as [59] (following a different experimental setup), and the aggregation is done by taking the mean of the metric across all subjects for every iteration.

5 Results and Discussion

To evaluate the flow network, we first report preliminary results as discussed in Section 4.3.1. As mentioned earlier, the performance measure for this experiment is the average EPE. We then show the results of the ablation study for the experiments trained on the full dataset, and compare the networks using the average EPE and AAE. Next, we evaluate a number of other popular optical flow methods on the test set. These include FlowNet2.0 [20], FlowNet3.0 [21], LiteFlowNet [23], PWC-Net [22], and the classic Gunnar-Farneback optical flow [60]. These CNN-based methods are chosen due to their impact and relevance in the community. To select Gunnar-Farneback flow, we used Allaert et al.'s [30] review, who recommend optical flow methods well-suited for facial motion, namely Gunnar-Farneback, Flow-Field, and Normalized Convolutional Upsampling (NCUP). Performance varied depending on the datasets, but Farneback, FlowField, and TV-L1 were consistently among the top. Based on this, we felt that Farneback would be an appropriate candidate from the non-CNN category.

Finally, the results of the micro-expression detection task using all networks are presented, which shows the usefulness of our method for a practical application. For this task, we use the same networks as in optical flow prediction with the addition of TV-L1, since the latter was used to compute the optical flow in STSTNet [8], the microexpression classifier we used in this work.

5.1 Results for Ablation Studies

We first describe the initial results of Exp. 1, which comprise of the networks trained and tested on faces, FlyingChairs, and Sintel datasets. We report the performance in terms of average EPE. The average EPE is computed for only the largest flow prediction, which is defined by the $k = 1$ term from Equation (4) and obtained by setting $w_1 = 1$. This represents the average EPE for the largest flow prediction. Table 1 shows these preliminary results as described in Section 4.3.1. It is worth noting that the subjects that appear in the training set do not appear in the validation or test sets of our face data.

The error values in Table 1 are in pixels, averaged over each of the test sets. Row 1 shows the results when the network was trained on faces and tested on all three datasets. Similarly, rows 2 and 3 are trained on FlyingChairs and Sintel and tested on all three. From Table 1, we observe that the network trained on our BP4D-derived face dataset performs best when tested on faces. This is likely due to the nature of the dataset the network was trained on. The flow fields on our face dataset consist of small, non-rigid motions, especially when the head motion is lacking, whilst the motion fields in the FlyingChairs dataset have larger magnitude and is more rigid. The Sintel dataset is also different in nature than the face dataset. It is likely that the network trained on FlyingChairs overestimates the motion on the face dataset. Note that the results in Table 1 are comparable to state of the art methods on the Sintel dataset, as can be seen in [48].

One interesting thing to note here is that the performance of the optical flow algorithm when trained on face data and tested on Sintel data, is much better compared to when trained on Sintel data and tested on Sintel data. We conjecture that this may show the usefulness of our generated dataset to problems that are even unrelated to faces. On a side note, the optical flow contained in the Sintel dataset is notable for its large motion and occlusions [48]. Due to the large flow vectors present in Sintel, we suspect that the network trained on Sintel tends to also predict flow fields with large vector magnitudes when tested on Sintel. Hence, in our FlownetS implementation, large erroneous predictions may have impacted the EPE more than the smaller-valued predictions from the face-trained model. It is also possible that the range of motions present in the face dataset trains the network for a variety of scenarios. However, the results for training and testing on Sintel dataset may be further improved by using a model better adapted to the optical flow challenges present in the Sintel dataset, but this may make the model more dataset specific.

TABLE 2
Flow Performance for the Ablation Studies

	BP4D-Spontaneous		CK+	
Experiments	Ave. EPE	AAE	Ave. EPE	AAE
Exp. 1	0.2856	0.1975	**0.2343**	**0.2080**
Exp. 2I	0.4610	0.3033	0.7936	0.4786
Exp. 2II	**0.2498**	**0.1728**	0.2821	0.2440
Exp. 3I	0.7010	0.4524	1.1573	0.5869
Exp. 3II	0.4660	0.2887	1.0082	0.4640

After adding the cyclic loss and training for more data and epochs, we expect to observe a difference in performance compared with Exp. 1. Here, we train the setup for Exp 1 again, using the same data split as the other experiments, for comparison purposes. Now we show the results of the networks trained with cyclic loss as described in Sections 4.3.2 and 4.3.3.

Table 2 summarizes the statistics computed based on the results of Exp. 2 and Exp. 3, evaluated on all 32.5k image pairs in the BP4D-Spontaneous test set and on 9930 out of a total of 10115 available consecutive image pairs of the CK+ dataset. The ground-truth for the image pairs from CK+ were computed in an identical manner to those in BP4D-Spontaneous, but some samples were omitted due to some errors in the generation process.

The average EPE is defined the same way as in Table 1, and the AAE is computed exactly as defined in Equation (8), since the loss function for the AAE is only evaluated on the largest flow prediction, contrary to the EPE. As outlined at the end of Section 4.3, Exp. 2I and Exp. 2II represent, respectively, the higher and lower reconstruction weight experiments, while Exp. 3I and Exp. 3II represent the experiment with smoothness constraint and the experiment with average angular error.

There are several observations to be made from these results. Adding the cyclic loss but with *lower* reconstruction weights (Exp. 2II) improves the flow prediction compared to using only the EPE loss (Exp. 1), since both EPE and AAE decrease significantly. When there is higher weight on reconstruction loss (Exp. 2I), the network alters the predicted flow to improve the warped output's semblance to X_2. However, the higher focus on reconstruction worsens the performance of the AAE and EPE. One reason could be that the noisy ground truth does not necessarily reconstruct X_2 from X_1 very well, i.e., the reconstruction capability of a predicted flow field is adversarial to the ground-truth flow EPE and AAE.

TABLE 3
Comparing Various Optical Flow Methods

	BP4D-Spontaneous		Improvement		CK+		Improvement	
Optical flow methods	Ave. EPE	AAE	Ave. EPE	AAE	Ave. EPE	AAE	Ave. EPE	AAE
Exp. 1 (this work)	0.2856	0.1975	12.54%	12.51%	**0.2343**	**0.2080**	0% (ref.)	0% (ref.)
Exp. 2II (this work)	**0.2498**	**0.1728**	0% (ref.)	0% (ref.)	0.2821	0.2440	16.94%	14.75%
PWC-Net	1.1538	0.4643	78.35%	62.78%	1.2340	0.7049	81.01%	70.49%
FlowNet2.0	0.6719	0.4347	62.82%	60.25%	0.6496	0.5078	63.93%	59.04%
FlowNet3.0-CSS	0.6839	0.4457	63.47%	61.23%	0.5168	0.4029	54.66%	48.37%
LiteFlowNet	0.7226	0.4771	65.43%	63.78%	0.6306	0.4826	62.84%	56.90%
Gunnar-Farneback	0.3670	0.2294	31.93%	24.67%	0.3837	0.3118	38.94%	33.29%

Exp. 3 with the smoothness and AAE losses yields worse outcomes than the other two in terms of predicted flow, particularly compared to Exp. 2I. Note that Exp. 3 weights are initialized from the latter to test any change in performance. This could be due to the decreased weight in the EPE loss, which suggests that the EPE is a stronger indicator of flow performance than the AAE. The EPE encodes the direction in addition to the magnitude information. Another explanation would be that training data with angular error as a loss metric does not generalize well to test data, unlike the EPE. Exp. 3I exhibits the worst performance in both EPE and AAE amongst our network variants. This is likely due to the imposed smoothness constraints, which impose flow field values in the otherwise null regions outside the face boundary.

The performance of the network trained on the BP4D-Spontaneous face data and tested on the CK+ data, reveals interesting insights. Indeed, for almost all experiments, there is a decrease in performance from BP4D-Spontaneous to CK+. This is less pronounced in terms of AAE performance. Note that this decrease in performance may be expected due to cross-database variations. In addition, even with the decrease, the performance is fairly decent and shows that the learnt knowledge is transferable to another dataset. This addresses any concern of database bias.

5.2 Comparison With Other Networks

We now compare the results with other notable optical flow implementations. Table 3 shows the flow statistics computed for the network variants described earlier.[3] The improvement is defined to be the percentage improvement of the best performing networks on BP4D-Spontaneous and CK+ test sets (which are Exp. 2II and Exp. 1 respectively), over the others.

In all cases, the networks trained on our automatic face dataset perform better in both metrics than PWC-Net [22] and LiteFlowNet [23], which are some of the popular CNN-based optical flow methods. PWC-Net demonstrates a notably high average EPE, but a more competitive AAE. This is likely due to an overestimation of the flow prediction magnitudes. FlowNet2.0 and FlowNet3.0-CSS, which are both state of the art improvements on FlowNetS, are both outperformed by all of our network variants with the exception of average EPE in Exp. 3I. The Gunnar-Farneback optical flow performs better than all methods in both average EPE and AAE, but is outperformed by Exp. 1 and Exp. 2II. These results confirm Farneback's high performance for facial motion as reported in [30], as it outperforms all of the other CNN-based works. It also helps confirm that our generated ground-truth is reliable, since the flow computed using Farneback is the closest flow field amongst the other network variants.

Our work (Exp. 1 and Exp. 2II) outperforms both the CNN-based methods and Gunnar-Farneback in both metrics, when tested on both BP4D-Spontaneous and CK+. Gunnar-Farneback is still the most accurate amongst the CNN-based methods. Note that in the case of CK+, we trained the networks on BP4D-Spontaneous data, which is a completely different dataset. This suggests that our method can be extended to face datasets that are different from those in the training set and still achieve superior performance than optical flow CNNs that were not trained on faces, indicating that the learnt knowledge is fairly independent of the dataset. An improvement would be to also include grayscale images in the training set, since their presence in CK+ is a possible reason for the decrease in performance seen by our networks.

To investigate the type of flow produced by each of the networks on the facial images, Fig. 4 shows a sample subset of image pairs in the test set with their respective ground-truth and flow predictions from each network. The EPE and AAE for each prediction are also labeled, computed the same way as in Tables 2 and 3. The saturation intensity in a given image is only representative of the intensity of that region relative to the other pixels of the same image. The same intensity in two images may have substantially different optical flow vector values. This is common practice in optical flow visualization, since it places emphasis on which motion is more salient for a given image. In images with small motion, as is the case in many frames in the BP4D dataset, using to-scale visualization would not convey important local motion information. We note that the following remarks for the remainder of this section are qualitative in nature and are based on a very small subset, but nevertheless yield some insight to accompany the statistics from Tables 2 and 3. We first observe the differences in flow predictions among the networks trained on our dataset. From these five, Exp. 1 shows the sparsest predictions, which is expected as it only minimizing the EPE from the sparse ground-truth flow. After introducing the cyclic loss in the other four experiments, denser optical features start to appear, caused by the added emphasis on image reconstruction. For example, this denser optical flow allowed the network to better predict the eye motion in rows 3 and 4 of

3. The interested reader is referred to the supplemental material, available at https://www.dropbox.com/s/o7158gi46tppvb1/SupplementalMaterial_OpticalFlow.docx?dl=0, for the error histograms for both ablation studies and comparison results.

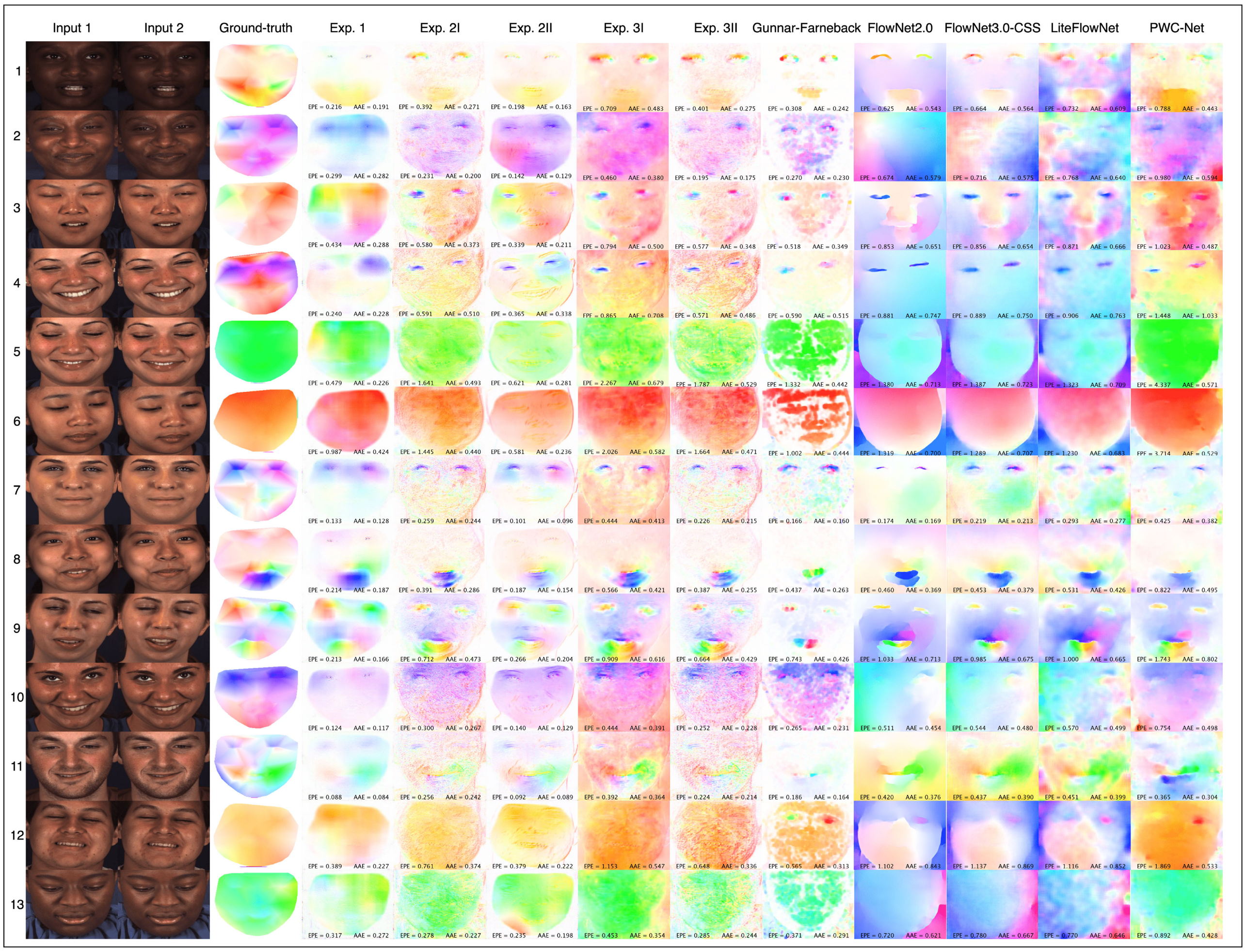

Fig. 4. Color-coded optical flow predictions for a small subset of the test set for the networks trained in each of the experimental setups. The examples contain different types of facial motion, meant to illustrate the type of flow outputs produced by each network for qualitative assessment.

Fig. 4. Thus, the EPE loss taught the network to predict well the regional directions and magnitudes, and the cyclic loss helped it further localize the motion in these regions.

There is higher motion variance across the face in Exp. 2I compared to Exp. 2II. Although both were trained with cyclic loss, there is higher emphasis on the loss in Exp. 2I than in Exp. 2II, which more clearly shows the effect of the cyclic loss, since no other losses were introduced in Exp. 2. The outputs of the networks with cyclic loss also show coarser representations compared to the outputs of FlowNet2.0 and FlowNet3.0-CSS, such as in rows 9 and 10 in Fig. 4. When the smoothness constraints were imposed in Exp. 3I, the face segmentation learned by the network was affected, since the large values of the flow derivatives at the face boundaries enlarged the gradients in the smoothness loss function.

By both visual perception of these examples and the average EPE and AAE values from Table 3, the Gunnar-Farneback optical flow shows similarity in both direction and magnitude. Since the method is unbiased by any training data, this similarity provides a degree of validation to the ground-truth optical flow. However, it still underestimates optical flow in some instances, such as the near-zero regions in rows 6, 8, and 13. The Gunnar-Farneback flow also segments the face, since the background has zero motion. This is in contrast to the outputs of the other four networks (FlowNet2.0, FlowNet3.0-CSS, LiteFlowNet, and PWC-Net). The outputs of FlowNet2.0, FlowNet3.0-CSS, and LiteFlowNet all tend to estimate background flow. The flow trend of their outputs from the examples of Fig. 4 can be matched with the outputs of the other networks, although some examples — especially those with global motion, such as in rows 4, 5, and 13 — show appreciable differences. PWC-Net demonstrates a more consistent flow pattern similar to our networks and Gunnar-Farneback. However, the EPE values in both the Fig. 4 examples and Table 3 suggest that the network, perhaps, overestimates the magnitude of the optical flow vectors in the field. Although its AAE is the second-highest, its value is close to several of the other methods. However, its EPE is significantly higher in comparison. This fact, complemented with the shown examples, suggests that the direction is a lesser problem than magnitude in PWC-Net.

The examples in rows 5, 6, 12, and 13 are characterized by predominantly global motion in one direction only. In these examples, the subject is mainly tilting their heads without any change in expression. Those in rows 4, 7, 8, 9 and 10 have the local motion as their salient feature, mainly in the eyes and mouth regions. Local face motion is more indicative of changes in facial expression, and the network's ability to identify the local motion can be used in FER. The remaining examples are rich with both global and local motions, indicating more aggressive motion along with the change in expressions. From these examples, all the networks were able to identify the local motions, except row 2, where three of the networks were not

TABLE 4
Details of the SMIC and SAMM Datasets Used for Micro-Expression Detection [8], [52], [53]

	No. of Samples per Class			Gender		Ethnicities	Mean age
Dataset	Positive	Negative	Surprise	Male	Female		
SAMM	26	92	15	16	16	13	33.2
SMIC	51	70	43	10	6	2	28.1
Combined	77	162	58	26	22	13	30.4

The number of samples per class is also the number of video clips.

TABLE 5
Results of the Aggregated Performance Metrics With Their Deltas for Micro-Expression Recognition on the **SAMM** and **SMIC** Datasets Separately, Using TVL1 Optical Flow as Done in [8], Our Network With Different Variants, and the Other Optical Flow CNN Architectures

Optical Flow Methods	SAMM					SMIC				
	P_M	R_M	F_{1_M}	F_{1_μ}	G_M	P_M	R_M	F_{1_M}	F_{1_μ}	G_M
TVL1	0.800, 0.088	**0.777, 0**	0.773, 0.021	0.698, 0.029	0.513, 0.015	0.612, 0.029	*0.606, 0.053*	0.589, 0.048	0.549, 0.058	0.385, 0.108
Exp. 1	**0.888, 0**	0.737, 0.040	**0.794, 0**	**0.727, 0**	0.429, 0.099	0.480, 0.161	0.518, 0.141	0.489, 0.148	0.404, 0.203	0.263, 0.230
Exp. 2I	0.763, 0.125	0.713, 0.064	0.722, 0.072	0.636, 0.091	0.411, 0.117	0.610, 0.031	0.585, 0.074	0.581, 0.056	0.518, 0.089	0.395, 0.098
Exp. 2II	0.849, 0.039	0.745, 0.032	0.784, 0.010	*0.713, 0.014*	0.487, 0.041	0.582, 0.059	0.543, 0.116	0.546, 0.091	0.432, 0.175	0.280, 0.213
Exp. 3I	0.765, 0.123	0.735, 0.042	0.738, 0.056	0.653, 0.074	0.457, 0.071	0.621, 0.020	0.591, 0.068	0.593, 0.044	0.538, 0.069	0.350, 0.143
Exp. 3II	*0.816, 0.072*	0.770, 0.007	*0.780, 0.014*	0.715, 0.012	0.516, 0.012	0.603, 0.038	0.580, 0.079	0.577, 0.060	0.514, 0.093	0.364, 0.129
Gunnar-Farneback	0.719, 0.169	0.696, 0.081	0.690, 0.104	0.608, 0.107	0.412, 0.116	0.570, 0.071	0.479, 0.180	0.505, 0.132	0.387, 0.220	0.217, 0.276
FlowNet2.0	0.724, 0.164	0.717, 0.06	0.695, 0.099	0.650, 0.077	0.431, 0.097	**0.641, 0**	0.627, 0.032	*0.622, 0.015*	*0.552, 0.055*	*0.420, 0.073*
FlowNet3.0-CSS	0.813, 0.075	*0.764, 0.013*	0.773, 0.021	0.711, 0.016	**0.528, 0**	*0.633, 0.008*	**0.659, 0**	0.636, 0.001	0.592, 0.015	**0.493, 0**
LiteFlowNet	0.767, 0.121	0.761, 0.016	0.749, 0.045	0.694, 0.033	*0.514, 0.014*	0.635, 0.006	0.658, 0.001	**0.637, 0**	**0.607, 0**	0.438, 0.055
PWC-Net	0.742, 0.146	0.738, 0.039	0.722, 0.072	0.676, 0.051	0.485, 0.044	0.501, 0.131	0.525, 0.134	0.501, 0.136	0.422, 0.185	0.249, 0.244

The best results in each column are in bold blue font, the second best are underlined, and the third best are in blue italic font.

able to pick up the eye movements. The differences in network outputs are clearer in the examples with global motion.

From the overall results of the experiments, one may note that our networks trained on the automatically generated face dataset are better-suited at predicting the optical flow on faces compared to other networks. Our model can be applied to predict optical flow on any sets of frontal / near frontal faces. The only pre-processing required is the detection of key-points using OpenFace [43], cropping the face using the maximum and minimum x-y coordinate values plus a small offset in each coordinate, and resizing the resulting image to a resolution of 512×384 pixels to prepare it as an input for FlowNetS. We use the offset values of 10 pixels in all four directions, for cropping.

5.3 Results of Micro-Expression Detection

We now report on the results of the experiments described in Section 4.4.2 for micro-expression detection. The details of the SAMM and SMIC datasets used for training and testing are given in Table 4 [52], [53], [8].

The micro- and macro-averaged metrics are shown for every network on each of the SAMM, SMIC, and combined datasets in Tables 5 and 6. In these tables, the aggregation of the metrics across the subjects from the LOSOCV is the mean of the metric across the subjects. The delta values are also computed, where the delta is defined to be the absolute value of the difference between a statistic and the maximum of that statistic in the column.

TABLE 6
Results of the Aggregated Performance Metrics With Their Deltas for Micro-Expression Recognition on the *combined SAMM and SMIC Dataset*, Using TVL1 Optical Flow as Done in [8], Our Network With Different Variants, and the Other Optical Flow CNN Architectures

Optical Flow Methods	SAMM and SMIC				
	P_M	R_M	F_{1_M}	F_{1_μ}	G_M
TVL1	*0.740, 0.0170*	*0.714, 0.013*	0.711, 0.002	0.662, 0.003	0.491, 0.020
Exp. 1	0.692, 0.065	0.601, 0.126	0.622, 0.091	0.548, 0.117	0.307 , 0.204
Exp. 2I	0.726, 0.031	0.700, 0.027	0.694, 0.091	0.635, 0.117	0.465, 0.046
Exp. 2II	**0.757, 0**	0.674, 0.053	0.702, 0.011	0.614, 0.051	0.342, 0.052
Exp. 3I	0.726 , 0.031	0.706 , 0.021	0.702 , 0.011	*0.661, 0.004*	0.459, 0.052
Exp. 3II	0.704, 0.053	0.678 , 0.049	0.676 , 0.037	0.592 , 0.073	0.447 , 0.064
Gunnar-Farneback	0.720, 0.037	0.700, 0.027	0.690, 0.023	0.629, 0.036	0.472, 0.039
FlowNet2.0	0.745, 0.012	0.709, 0.018	*0.709, 0.004*	0.636, 0.029	0.459, 0.052
FlowNet3.0-CSS	0.725, 0.032	**0.727, 0**	**0.713, 0**	**0.665, 0**	**0.511, 0**
LiteFlowNet	0.717, 0.040	0.717, 0.010	0.702, 0.011	0.650, 0.015	0.446, 0.065
PWC-Net	0.704, 0.053	0.693, 0.034	0.687, 0.026	0.631, 0.034	*0.482, 0.029*

The best results in each column are in bold blue font, the second best are underlined, and the third best are in blue italic font.

The results in Table 5 indicate that the performance of STSTNet trained on optical flow features from different networks also significantly depends on the dataset it is trained on. For each evaluation measure, the top three performing networks are indicated in bold blue, underlined, and italic fonts respectively.

We note that the F_1-scores are typically lower than the precision and recall since these are aggregated metrics, i.e., the F_1-score averaged over all F_1-scores in the LOSOCV, and is not the harmonic mean of the aggregated precision and recall.

For macro-averaged precision P_M, as well as the macro and micro-averaged F_1-scores on the SAMM dataset, the top scores are achieved from Experiments 1, 2II, and 3II, followed closely by FlowNet3.0-CSS and TVL1. The higher F_1-scores are more influenced by the precision values and less so by the recalls. Exp. 1 is the highest for these three metrics, while TVL1 scored highest in R_M, and FlowNet3.0-CSS in geometric mean. This is one testimony to the complexity of capturing the overall classification performance with a single scalar metric for multi-class problems, since the proposed metrics can each emphasize different features of the classifier performances. Exp. 3II is the only variant which is consistently among the top 3 for all metrics, at either second or third.

The SMIC results allow for a more consistent inference on the performance of the classifiers. Across all metrics, the top three networks were FlowNet2.0, FlowNet3.0-CSS, and LiteFlowNet. Both the precision and recall, and consequently the F_1-scores, follow more similar trends, in contrast with the SAMM and combined training protocols. For precision, recall, and F_1-scores, the lowest three scores are interchanged amongst Exp. 1, PWC-Net, and Gunnar-Farneback. In fact, Gunnar-Farneback and PWC-Net are consistently the least performing across all three training protocols.

By comparing the results across the three training protocols, it is difficult to conclude that optical flow features computed from one specific method will be optimal for training the STSTNet classifier for micro-expression detection. Although the networks trained using our method performed well when trained and tested on SAMM, they were somewhat outperformed in the other two protocols. However, even in these cases, they were not as consistently behind when compared to Gunnar-Farneback and PWC-Net, which can be seen by the delta values in Tables 5 and 6. This could be due to the sparse nature of the learned optical flow representations from our generated dataset. It is also plausible that the accuracy of the flow magnitude prediction may not be a consistent predictor of its performance on micro-expression detection. What we mean here, is not the magnitude in general, but rather magnitude in regions that do not correspond or assist in microexpression detection. For example, large head motion will have large error magnitudes associated with it. However, smaller regions in the same face such as eyes or mouth may have smaller error magnitudes that may be key for a microexpression. Hence, a good global EPE statistic can be a result of the (correct) detection of the head motion despite a relatively less accurate estimate of the motion in the mouth/eye region.

We hypothesize that our method will overcome the performance difference in some of the results if we use a denser keypoint tracker during the optical flow training phase to generate the BP4D ground-truth. This will likely improve the network's ability to more consistently capture fine local facial motion which may otherwise have been missed in the current work. Furthermore, as previously discussed, we have used FlowNetS to train the face data to benchmark its efficacy compared to other networks, and thus using a better-designed CNN along with the denser keypoint ground-truth will likely further improve the performance.

6 Conclusion and Future Work

In this paper, we explore the possibility of using a facial expression dataset to learn optical flow representations based on a self-supervised technique. Motion information on faces has been shown to be useful in facial expression analysis in multi-modal techniques.

The dataset is generated by using the image sequences from the BP4D-Spontaneous dataset to compute the optical flow ground-truth. The OpenFace 2.0 toolbox, which uses a constrained local model, is used to locate the facial landmarks on every image. Delaunay triangulation is then used on the resulting set of points to form the face mesh and allow the computation of the optical flow for every pair of images using triangle-to-triangle affine maps to develop an automatic facial optical flow dataset. The generated dataset, with a total of nearly 324k image pairs, is used as a *noisy* ground-truth for optical flow to train the FlowNetS convolutional autoencoder architecture with 228k pairs in the training partition.

It was observed that training the FlowNetS architecture for optical flow on this automatically generated noisy ground-truth data improved the network's ability to predict optical flow on face data in particular. The learned representations also helped the network give good accuracy on the Flying-Chairs and Sintel datasets. This demonstrates that the facial movements are nicely encoded in our data which enables the network to learn subtle movements that are useful on the challenging Sintel dataset as well. A cyclic loss was also added for optimization to help the network use the predicted flow to reconstruct the second image, and the flow results from different experimental setups are compared. It was seen that the flow predictions are best when there is less emphasis on reconstruction, due to denser representations learned with reconstruction that are not present in the ground-truth flow fields. Compared with other optical flow methods (Gunnar-Farneback, FlowNet2.0, FlowNet3.0-CSS, LiteFlowNet, and PWC-Net), it was shown that the networks trained on the generated dataset predict better flow representations, as quantified by the flow error metrics. This implies that a network trained on good face optical flow ground-truth have the propensity to outperform networks trained on other datasets.

To investigate the performance of the different optical flow network variants in an FER application, the optical flow features were used to train STSTNet for micro-expression detection. The experimental results using different performance metrics were mixed, e.g., FlowNet3.0-CSS performed better in a number of metrics. However, our method also demonstrated promising results in a number of cases, particularly those on the SAMM dataset. Note that further improvements and extension to this baseline work can help improve its application to FER.

For further investigation and improvement, future work related to this work can include the following:

1) Use a denser tracker such as Zface [61] to track a higher number of key-points for a finer triangulation and denser optical flow ground-truth in our automatic data generation algorithm.
2) Use a more complex CNN architecture to train the denser optical flow ground-truth.
3) Train the optical flow network on faces with some head rotation, such as pan and tilt, to learn optical flow for non-frontal faces.
4) Tackle challenges in optical flow learning, such as in environments with occlusion and illumination, to increase the robustness of facial optical flow.

In addition to these improvements for optical flow learning, the empirical analysis can be extended to evaluate the performance of the face-trained optical flow CNN in other problems in facial expression analysis, such as action unit recognition.

ACKNOWLEDGMENTS

We thank Ms. Florence Wacheux for her input in the review of the manuscript draft.

REFERENCES

[1] S. Söderkvist, K. Ohlén, and U. Dimberg, "How the experience of emotion is modulated by facial feedback," *J. Nonverbal Behav.*, vol. 42, no. 1, pp. 129–151, Mar. 2018.
[2] C. F. Benitez-Quiroz, R. B. Wilbur, and A. M. Martinez, "The not face: A grammaticalization of facial expressions of emotion," *Cognition*, vol. 150, pp. 77–84, May 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0010027716300324
[3] S. M. Gillespie, P. Rotshtein, L. J. Wells, A. R. Beech, and I. J. Mitchell, "Psychopathic traits are associated with reduced attention to the eyes of emotional faces among adult male non-offenders," *Front. Hum. Neurosci.*, vol. 9, Oct. 2015, Art. no. 552. [Online]. Available: https://www.frontiersin.org/article/10.3389/fnhum.2015.00552
[4] A. J. O'Toole, *Psychological and Neural Perspectives on Human Face Recognition*, Berlin, Germany: Springer, 2005, pp. 349–369.
[5] X. Zhang et al., "BP4D-spontaneous: A high-resolution spontaneous 3D dynamic facial expression database," *Image Vis. Comput.*, vol. 32, pp. 692–706, Jun. 2014.
[6] A. Dosovitskiy et al., "FlowNet: Learning optical flow with convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 2758–2766. [Online]. Available: http://lmb.informatik.uni-freiburg.de/Publications/2015/DFIB15
[7] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, "The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops*, 2010, pp. 94–101.
[8] S. Liong, Y. S. Gan, J. See, H. Khor, and Y. Huang, "Shallow triple stream three-dimensional cnn (ststnet) for micro-expression recognition," in *Proc. 14th IEEE Int. Conf. Autom. Face Gesture Recognit.*, 2019, pp. 1–5.
[9] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. 7th Int. Joint Conf. Artifical Intell.*, 1981, pp. 674–679. [Online]. Available: http://dl.acm.org/citation.cfm?id=1623264.1623280
[10] M. R. BalazadehBahar and G. Karimian, "High performance implementation of the horn and schunck optical flow algorithm on FPGA," in *Proc. 20th Iranian Conf. Elect. Eng.*, 2012, pp. 736–741.
[11] S. Shah and X. Xuezhi, "Traditional and modern strategies for optical flow: An investigation," *SN Appl. Sci.*, vol. 3, pp. 1–14, 2021.
[12] J. Janai, F. Güney, A. Ranjan, M. Black, and A. Geiger, "Unsupervised learning of multi-frame optical flow with occlusions," in *Proc. 15th Eur. Conf. Comput. Vis.*, 2018, pp. 713–731.
[13] S. Meister, J. Hur, and S. Roth, "UnFlow: Unsupervised learning of optical flow with a bidirectional census loss," *Proc. AAAI Conf. Art. Intell.*, vol. 32, no. 1, pp. 7251–7259, 2018.
[14] Z. Ren, J. Yan, X. Yang, A. Yuille, and H. Zha, "Unsupervised learning of optical flow with patch consistency and occlusion estimation," *Pattern Recognit.*, vol. 103, 2020, Art. no. 107191. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320319304911
[15] D. Sun, S. Roth, and M. J. Black, "Secrets of optical flow estimation and their principles," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 2432–2439.
[16] L. Tian, Z. Tu, D. Zhang, J. Liu, B. Li, and J. Yuan, "Unsupervised learning of optical flow with CNN-based non-local filtering," *IEEE Trans. Image Process.*, vol. 29, pp. 8429–8442, 2020.
[17] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7794–7803.
[18] Y. Niu, A. Dick, and M. Brooks, "Discontinuity-preserving optical flow computation by a dynamic overdetermined system," in *Proc. 9th Biennial Conf. Aust. Pattern Recognit. Soc. Digit. Image Comput. Techn. Appl.*, 2007, pp. 352–359.
[19] L. Alvarez, R. Deriche, T. Papadopoulo, and J. Snchez, "Symmetrical dense optical flow estimation with occlusions detection," *Int. J. Comput. Vis.*, vol. 75, no. 3, pp. 371–385, Dec. 2007.
[20] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "FlowNet 2.0: Evolution of optical flow estimation with deep networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1647–1655. [Online]. Available: http://lmb.informatik.uni-freiburg.de/Publications/2017/IMSKDB17
[21] E. Ilg, T. Saikia, M. Keuper, and T. Brox, "Occlusions, motion and depth boundaries with a generic network for disparity, optical flow or scene flow estimation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 626–643. [Online]. Available: http://lmb.informatik.uni-freiburg.de/Publications/2018/ISKB18
[22] D. Sun, X. Yang, M. Liu, and J. Kautz, "PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8934−8943.
[23] T.-W. Hui, X. Tang, and C. C. Loy, "LiteFlowNet: A lightweight convolutional neural network for optical flow estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8981–8989. [Online]. Available: http://mmlab.ie.cuhk.edu.hk/projects/LiteFlowNet/

[24] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Intl. Conf. Computer Vis.*, 2017, pp. 2242–2251.

[25] J. J. Yu, A. W. Harley, and K. G. Derpanis, "Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 3–10.

[26] W.-S. Lai, J.-B. Huang, and M.-H. Yang, "Semi-supervised learning for optical flow with generative adversarial networks," in *Proc. 31st Int. Conf. Neural Informat. Process. Syst.*, 2017, pp. 353–363.

[27] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," *Adv. Neural Info. Process. Sys.*, vol. 28, 2015.

[28] P. Snape, A. Roussos, Y. Panagakis, and S. Zafeiriou, "Face flow," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 2993–3001.

[29] H. Le, T. Nimbhorkar, T. Mensink, A. S. Baslamisli, S. Karaoglu, and T. Gevers, "Unsupervised generation of optical flow datasets from videos in the wild," 2018, *arXiv:1812.01946*.

[30] B. Allaert, I. R. Ward, I. M. Bilasco, C. Djeraba, and M. Bennamoun, "Optical flow techniques for facial expression analysis: Performance evaluation and improvements," 2019, *arXiv: 1904.11592*.

[31] M. R. Koujan, A. Roussos, and S. Zafeiriou, "Deepfaceflow: In-the-wild dense 3D facial motion estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 6617–6626.

[32] C. Ma, L. Chen, and J. Yong, "Au r-cnn: Encoding expert prior knowledge into r-cnn for action unit detection," *Neurocomputing*, vol. 355, pp. 35–47, Aug. 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231219305338

[33] H. Yang and L. Yin, "Learning temporal information from a single image for au detection," in *Proc. 14th IEEE Int. Conf. Autom. Face Gesture Recognit.*, 2019, pp. 1–8.

[34] Y. Li, J. Zeng, S. Shan, and X. Chen, "Self-supervised representation learning from videos for facial action unit detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 10924–10933.

[35] A. Romero, J. León, and P. Arbeláez, "Multi-view dynamic facial action unit detection," *Image Vis. Comput.*, vol. 122, 2018, Art. no. 103723. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0262885618301598

[36] W. Chu, F. De la Torre, and J. F. Cohn, "Learning spatial and temporal cues for multi-label facial action unit detection," in *Proc. 12th IEEE Int. Conf. Autom. Face Gesture Recognit.*, 2017, pp. 25–32.

[37] N. Perveen, D. Roy, and C. K. Mohan, "Spontaneous expression recognition using universal attribute model," *IEEE Trans. Image Process.*, vol. 27, no. 11, pp. 5575–5584, Nov. 2018.

[38] Y. Gan, S.-T. Liong, W.-C. Yau, Y.-C. Huang, and L.-K. Tan, "OffApexNet on micro-expression recognition system," *Signal Process. Image Commun.*, vol. 74, pp. 129–139, May 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0923596518310038

[39] M. Verburg and V. Menkovski, "Micro-expression detection in long videos using optical flow and recurrent neural networks," in *Proc. 14th IEEE Int. Conf. Autom. Face Gesture Recognit.*, 2019, pp. 1–6.

[40] Q. Li, J. Yu, T. Kurihara, and S. Zhan, "Micro-expression analysis by fusing deep convolutional neural network and optical flow," in *Proc. 5th Int. Conf. Control, Decis. Informat. Technol.*, 2018, pp. 265–270.

[41] X. Wang and A. Gupta, "Unsupervised learning of visual representations using videos," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 2794–2802.

[42] N. Dubey, S. Ghosh, and A. Dhall, "Unsupervised learning of eye gaze representation from the web," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2019, pp. 1–7.

[43] T. Baltrusaitis, P. Robinson, and L. Morency, "Constrained local neural fields for robust facial landmark detection in the wild," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2013, pp. 354–361.

[44] A. Zadeh, T. Baltrusaitis, and L. Morency, "Deep constrained local models for facial landmark detection," 2016, *arXiv:1611.08657*.

[45] J. Gallier, *Geometric Methods and Applications: For Computer Science and Engineering*, 2nd ed., Berlin, Germany: Springer, 2013.

[46] S. Yan, Z. Zhang, Y. Fu, Y. Hu, J. Tu, and T. Huang, "Learning a person-independent representation for precise 3D pose estimation," in *Multimodal Technologies for Perception of Humans*, Berlin, Germany: Springer, 2007, pp. 297–306.

[47] C. Ericson, *Real-Time Collision Detection*. Boca Raton, FL, USA: CRC Press, 2004.

[48] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A naturalistic open source movie for optical flow evaluation," in *Proc. 12th Eur. Conf. Comput. Vis.*, 2012, pp. 611–625.

[49] P. J. Huber, "Robust estimation of a location parameter," *Ann. Math. Statist.*, vol. 35, no. 1, pp. 73–101, Mar. 1964.

[50] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," *Int. J. Comput. Vis.*, vol. 92, no. 1, pp. 1–31, Mar. 2011.

[51] W.-J. Yan et al., "Casme II: An improved spontaneous micro-expression database and the baseline evaluation," *PLoS One*, vol. 9, 2014, Art. no. e86041.

[52] A. K. Davison, C. Lansley, N. Costen, K. Tan, and M. H. Yap, "SAMM: A spontaneous micro-facial movement dataset," *IEEE Trans. Affec. Comput.*, vol. 9, no. 1, pp. 116–129, Jan./Mar. 2018.

[53] X. Li, T. Pfister, X. Huang, G. Zhao, and M. Pietikäinen, "A spontaneous micro-expression database: Inducement, collection and baseline," in *Proc. 10th IEEE Int. Conf. Workshops Autom. Face Gesture Recognit.*, 2013, pp. 1–6.

[54] S. Liong, J. See, K. Wong, A. C. Le Ngo, Y. Oh, and R. Phan, "Automatic apex frame spotting in micro-expression database," in *Proc. 3rd IAPR Asian Conf. Pattern Recognit.*, 2015, pp. 665–669.

[55] D. E. King, "Dlib-ml: A machine learning toolkit," *J. Mach. Learn. Res.*, vol. 10, pp. 1755–1758, 2009.

[56] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1867–1874.

[57] M. H. Yap, J. See, X. Hong, and S. Wang, "Facial micro-expressions grand challenge 2018 summary," in *Proc. 13th IEEE Int. Conf. Autom. Face Gesture Recognit.*, 2018, pp. 675–678.

[58] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Informat. Process. Manage.*, vol. 45, no. 4, pp. 427–437, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0306457309000259

[59] D. Gholamiangonabadi, N. Kiselov, and K. Grolinger, "Deep neural networks for human activity recognition with wearable sensors: Leave-one-subject-out cross-validation for model selection," *IEEE Access*, vol. 8, pp. 133 982–133 994, 2020.

[60] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," in *Proc. 13th Scand. Conf. Image Anal.*, 2003, pp. 363–370.

[61] L. A. Jeni, J. F. Cohn, and T. Kanade, "Dense 3D face alignment from 2D video for real-time use," *Image Vis. Comput.*, vol. 58, pp. 13–24, 2017. [Online]. Available: http://zface.org

Muhannad Alkaddour received the MS degree from the Mechatronics Engineering Graduate Program, the American University of Sharjah, in 2020. His research experience and interests include artificial intelligence, with emphasis on deep learning and computer vision, as well as robotics, control systems, and dynamical systems.

Usman Tariq (Member, IEEE) received the PhD degrees in electrical and computer engineering from the University of Illinois at Urbana-Champaign, in 2013. He is a faculty member with the Department of Electrical Engineering, American University of Sharjah, UAE. He has also worked as a research scientist with Computer Vision Group, Xerox Research Center Europe, France. His research interests include computer vision and affective computing.

Abhinav Dhall (Member, IEEE) received the PhD degree in computer science from the Australian National University, Canberra, Australia, in 2014. He currently leads the Centre for Applied Research in Data Sciences, Indian Institute of Technology Ropar, India, and also an adjunct senior lecturer with Monash University, Australia. His research interests include computer vision and affective computing. He was awarded the Best Doctoral Paper Award for ACM ICMR 2013, Best Student Paper Honourable mention for IEEE 1372AFGR, and Best Paper Nomination for IEEE ICME 2012. He is an associate editor of the *IEEE Transactions on Affective Computing*.