Grenze International Journal of Engineering and Technology, June Issue

Motion Estimation using Optical Flow Through A CNN based Pyramidal Warping and Cost Volume Approach: An Optimized PWC-Net Model

K K Kavitha[1] and Dr Rekha B Venkatapur[2]
[1]Assistant Professor, Dept. of CSE, Nitte Meenakshi Institute of Technology, Bengaluru, India
kavitha.kk@nmit.ac.in
[2]Prof. and Head, Dept. of CSE, K S Institute of Technology, Bengaluru, India
rekhabvenkatapur@ksit.edu.in

***Abstract*— The use of video in our everyday digital interactions is on the rise. With the advancement of higher-resolution contentanalysis and displays, the substantial volume of video content presents considerable challenges in terms of acquisition, trans- mission, compression, and display while maintaining high quality.Video compression is very crucial to manage the resources aswell as ensure smooth transmission and playback of digital videos across different platforms and devices. In our work, we introduce an optimized PWC-Net (Pyramidal Warping Cost Volume-Net) architecture for motion estimation, which is a computationally intensive initial component present in codecs forvideo compression, such as H.264 and H.265/HEVC. Proposed approach utilizes a CNN (Convolutional Neural Network) based PWC-Net architecture to estimate the optical flow between the frame sequences, which is considered as the motion information. Our model, compresses the video sequences efficiently without computing the motion vectors unlike the traditional motion es- timation techniques using block-based methods. It demonstrates its effectiveness with the experimental results when performing compression on various H.264 codecs. It learns efficient video compression without the need for motion computation unlike the traditional motion estimation techniques and our experiments show that an optimized PWC-Net outperforms good compressionratio by performing motion estimation through optical flow whencompared with existing motion estimation schemes on H.264 and H.265 codec.**

***Index Terms*— Video compression, Motion estimation, CNN, Optical flow, Optimized PWC-Net.**

I. INTRODUCTION

Motion Estimation occupies a significant portion in the process of video compression. It eliminates the temporal redundancy in video frames through inter-frame prediction. Previously, motion estimation was performed using a conventional block-based motion estimation algorithm to estimate the motion between consecutive frames. Specifically, in order to provide motion compensated prediction, each frame is dividedinto macroblocks and their motion vectors are determinedusing a block matching technique. Later this data is harnessed to minimize redundancy and encode the video data with optimal efficiency. Techniques for motion estimation determine pixel or block displacements between frames, empoweringthe encoder to express motion as vectors. These motion vectors, coupled with residual data are subsequently compressed,transmitted or stored. Obtaining the motion

Grenze ID: 01.GIJET.10.2.737

vector of eachmacroblock requires carrying out a thorough search using analgorithm that uses a block-matching criterion known as the*cost function* to assess every potential displacement insidea search frame. Full search is not practicable for real-timeimplementation due to its enormous computing cost, even if ithas the extra benefit of discovering the absolute best answer.The computational intricacy of a motion estimation methodcan be assessed through three key elements namely, the searchalgorithm, the cost function/the evaluation function and thesearch range parameter. Reducing the complexity of the chosencost function and/or the applied search technique will help to lower the complexity of motion estimating algorithms.However, the efficiency and quality of video compressionmay be affected by this decrease in complexity; hence, themotion estimation method is vital for keeping video qualityand deciding how efficient the video compression process is.

Conventional video compression standards such as MPEG- 4/H.264, HEVC and the latest VCC/H.266 utilize the seriesof predefined software modules having local decorrelatingdecompositions and block-based motion estimates such as the Discrete Cosine Transform (DCT) to process videos. Videocompression techniques have evolved significantly, movingfrom hand-designed algorithms to more advanced methodsthat utilize machine learning and Deep Neural Networks(DNN) for improved video compression efficiency [1]. Byleveraging deep neural networks, we can explore pixel-wisemotion estimation techniques to enhance video compression performance. Utilization of advanced methods like hybrid deeplearning model helps to attain a higher compression ratio,minimize latency and maintain the original video quality.Deep learning models are considered as extremely robust toolsfor creating optimal alternative video codecs due to theirexceptional capacity to acquire effective visual representations.A multitude of image compression architecture based on deeplearning methods have been put forth as a result [2]. The novelmodels have utilized Convolutional Neural Networks (CNN),Recurrent Neural networks (RNN), Generative Adversarial Networks (GAN) and autoencoders resulting in rate-distortion efficiencies that are purportedly on par with conventional image compression codecs such as JPEG and JPEG 2000. By the motivation of such developments, numerous researchers have developed deep video compression models that illustrate the significant potential of this overarching approach. The firstend-to-end learned deep video codec was presented by Wu et al. [3], using a hierarchical frame interpolation approach. A deep video compression codec based on blocks was described by Chen et al. [4]. An end-to-end video compression model (DVC) was developed by Lu et al. [5]. In this model, every component of a traditional hybrid video codec is swapped out with a deep learning model that has been trained as a whole asa hybrid architecture under a single loss function. Furthermore,a hierarchical video compression framework known as HLVC (Hierarchical Video Compression) was presented by Yang et al. [6].

In contemporary deep learning-based methods as well as traditional video codecs, a substantial share of system resources has been dedicated to motion estimation because motion estimation necessitates a resource-intensive search procedure, which we circumvent by training the network to adeptly depictthe disparities between the present frame and an array ofspatially-shifted neighboring frames. Though, deep learning- based methods have dominated this space, offering impressive results in terms of compression efficiency and quality. However, there is a growing interest in leveraging opticalflow technology as an alternative approach to perform motion estimation. This transition is driven by several factors, including the need for more computationally efficient solutions, robustness to various types of motion, and the desire to reducereliance on large-scale training datasets. This paper explores the evolution from deep learning methods towards optical flowtechnology for motion estimation in video compression.

In our work, we propose a network called an optimized PWC-Net to estimate the motion between the frame sequences.The PWC-Net [7] is a CNN based model designed for estimating motion between the video frame sequences usingoptical flow technology that avoids the computing cost ofexisting motion estimation techniques. Optical flow refers to the apparent motion of objects in an image or a sequenceof images. It computes the velocity of points within the images, offering an estimate of the potential locations of thesepoints in the subsequent image sequence. This technique is typically employed on a sequence of images with minimal temporal intervals, commonly found in video frames. Proposed network, an optimized PWC-Net is meticulously crafted from straightforward and firmly established principles-Pyramidalprocessing, Warping and Cost volume processing, the structureof which is shown below. Figure 1 depicts the network topology of PWC-Net from a general viewpoint. PWC-Net startsby using the input frames to create a feature pyramid. The network compares a pixel's features in the first picture with similar features in the second image at the top of the pyramid to generate a cost volume. Since this level has a lower spatial resolution, the cost volume is constructed using a compact range. In order to estimate the flow at the highest level, the aforementioned cost volume is subsequently provided as inputto a CNN together with characteristics from the initial picture.PWC-Net then uses rescaling and up sampling to project the predicted flow to the next level. PWC-Net uses the upsampledflow to warp the features of the second picture towards thefirst image, moving to the second-to-top pyramid level. This warping makes up for large motion,

therefore it's possible to use a narrower search range when building the cost volume. This new cost volume is then sent into a CNN for flow estimate at the current level, together with features from the initial picture and the previously unsampled flow. This level's estimated flow is then upsampled to the next (third) level, and so on, repeatedly, until the target level is reached. Thus PWC- Net is particularly effective in estimating dense optical flow fields that introduces noteworthy enhancements in terms of accuracy as well as model size in comparison to earlier CNN models designed for optical flow.

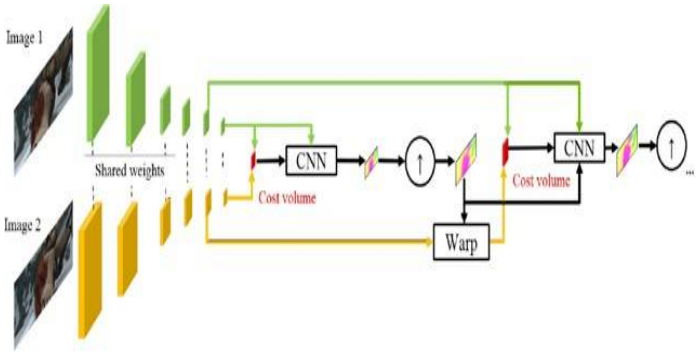

Figure 1. The structure of PWC-Net architecture.

II. Related Work

Optical flow estimation stands as a fundamental challenge within the realm of computer vision, finding diverse applications such as action recognition, autonomous driving and videoediting. Extensive research spanning decades has resulted in remarkable achievements in addressing demanding bench- marks [8][9][10]. The majority of the leading methods adhere to the energy minimization strategy that was first put out by Horn and Schunck [11]. However, fine-tuning complex energyfunctions usually turns out to be computationally intensive, rendering them less suitable for real-time applications. Adopt-ing convolutional neural networks' (CNNs') quick, scalable, and end-to-end trainable architecture is a very promising path.In recent years, this method has greatly advanced the area of computer vision.

Optical Flow. Variational techniques have been the dominant approach for estimating optical flow since Horn and Schunck's pioneering work [11, 16, 17]. This approach in- volves connecting the presumptions about brightness consistency and smoothness in space using an energy function. Over time, several enhancements have been introduced [12]- [15]. Recent efforts have focused on addressing large displacements, incorporating combinatorial matching into variational framework [16, 17]. Notably, the work presented in [17], referred to as Deep-Matching and Deep Flow, aggregates feature information from coarse-to-fine using convolutions and max-pooling but doesn't involve any learning and allparameters are configured manually. Subsequent work, like Epic Flow [18] has placed a strong emphasis on improving sparse matching quality, particularly when transitioning from sparse to thick flow fields while maintaining adherence to picture borders. Several researchers have previously explored theapplication of machine learning to optical flow. For instance, while Rosenbaum et al. [19] used Gaussian mixture models to simulate local optical flow statistics, Sun et al. [20] analyzed optical flow statistics and trained regularizes using Gaussian scale mixtures. Through a linear combination of these basic flows, Black et al. [21] predicted optical flow by computing the major components of a training set of flow fields. Other techniques used classifiers to predict occlusion probability [22]or choose between several inertial estimations [23]. Addition- ally, employing neural network models, work has been done onunsupervised learning of motion or disparity between frames in films. In order to represent connections between picture pairs, these approaches usually use multiplicative interactions,which allow for the interference of disparities and opticalflow from latent variables. A synchronous autoencoder is a kind of specialized autoencoder that Konda and Memisevic [24] utilized, whereas Taylor et al. [25] used factored gated limited Boltzman machines to tackle this problem. While these techniques are helpful for activity identification in moviesand have shown promise in controlled environments, when applied to real-world recordings, they are not more effective than traditional methods.

Early Efforts in Learning Optical Flow. Prior to the advent of deep learning, there exists a substantial history of research in learning optical flow. Here are some notewor thy contributions, Simoncelli and Adelson [26] delved into examining data matching problems related to optical flowin detail. Using examples from the synthetic blob world, Freeman et al. [27] concentrated on parameter learning inside the Markov Random field (MRF) model for picture motion. Roth and Black [28] used depth map-generated sequencesto study the spatial

statistics of optical flow. The learning method for the complete model of optical flow by Sun et al. [20] was restricted to a few training sessions [29]. Stochastic optimization methods were used by Li and Huttenlocker [30] to adjust parameters for the Black and Anandan approach [31].Nevertheless, there were only a limited number of factors that could be learned. Wulff and Black [32] pursued the learningof Principal Component Analysis (PCA) motion bases derived from optical flow estimates generated by GPUFlow using real movie data. While their approach was speedy, it tended to produce excessively smoothed flow results.

Novel Progress in the Study of Optical Flow. In response to the remarkable achievements in high- level vision tasks, thetwo CNN models, FlowNetS and FlowNetC, that Dosovitskiy et al. [12] developed, marked a significant shift in the optical flow estimation paradigm. Their research showed that a generic U-Net CNN architecture may be used to directly estimate optical flow from raw images. FlowNetS and FlowNetC stood out as the most advanced real-time techniques of their day, despite their performance being somewhat behind the state-of-the-art techniques. Mayer et al. [33] expanded the use of the FlowNet architecture to include tasks related to scene flow and disparity estimation. A unique strategy was used by Ilg et al. [34], who combined many simple FlowNet models into a more complex model. FlowNet2, which performed as well as or better than the most advanced benchmarks, especially on the Sintel test. Another advancement was the introduction of SpyNet, a compact spatial pyramid network,by Ranjan and Black [35]. On the Sintel test, SpyNet's performance was equivalent to that of the FlowNetC model showcasing its effectiveness even though it did not reach the absolute state-of-art performance.

CNN Models for Computer Vision's Dense Prediction Tasks. In computer vision, the denoising autoencoder [36]has emerged as a commonly employed technique used fordense prediction applications. It is particularly effective when combined with skip connections [37] linking the encoder and decoder, facilitating the prediction of dense outputs. Recent advancements, as seen in recent work [38] have highlighted the benefits of dilated convolution layers. These layers are adept atleveraging contextual information and enhancing fine-grained details, particularly in tasks like semantic segmentation. In the context of optical flow, we incorporate dilated convolutionsto harness contextual information, resulting in a notable improvement in performance. Another architectural innovation isthe DenseNet architecture [39, 40], which directly connects every layer to each other in a feedforward manner. This architecture has demonstrated superior accuracy and ease of training compared to traditional CNN layers, especially inimage classification tasks. We explore the application of the DenseNet concept in the context of dense prediction tasks for optical flow, aiming to further enhance performance.

III. Proposed Methodology

This section gives the detailed description of the proposed framework for the motion estimation phase in video compression. In this segment we explain the PWC-network to generateoptical flow and PWC-Net's fundamental components, Contextnetwork to post process and compress the generated optical flow.

A. The PWC-Net for Motion Estimation

In place of employing traditional block-based motion estimation, which estimates the motion between the current frame (y_t) and the previous reconstructed frame (y_{i-1}), the proposed model utilizes a CNN-based optimized PWC-Network [7] to generate the optical flow which is regarded as the motion information v_t. PWC-Net make use of a feature pyramid to estimate the optical flow between two adjacent frames and refines at a single pyramid level. PWC-Net performs feature wraping on the second image using unsampled flow, constructs a cost volume and then processes the cost volume using CNNs. More precisely the input to PWC Net consists of the previous frame x_t-1 and the current frame x_t, resulting in the optical flow v_t. This output is then compressed by the motion compression module. Moreover, unlike the compression method for motion information in traditional codecs, we propose a CNN based feed-forward context network to post process the flow explained below in section B.

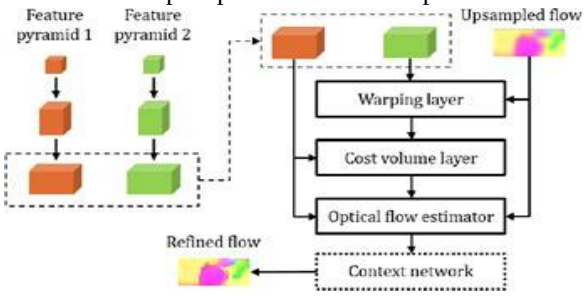

Figure 2. The components of the PWC-Net

Figure 2 shows the structural components of the PWC-network. Firstly, raw images are substituted by learnable feature pyramids in order to get fixed picture pyramid as raw photos are not affected by shadows or illumination variations [14]. Secondly, the warping operation from the conventional method is used to estimate vast motion. Thirdly, the network consists a layer to create the cost volume, which is then processed by CNN layers to estimate the flow, as the cost volume is a more discriminative representation of the optical flow than raw pictures. The model size is decreased by the costvolume and warping layers, which have no learnable parameters. Lastly, PWC-Net post-processes the optical flow usinga context network to improve the optical flow in contrast to using the contextual information by older standard techniques such as median filtering and bilateral filtering. The underlying mechanism of each component is discussed below in detail.

Pyramid Extractor Net for Features From two input frames, I_1 and I_2, L-level pyramids of feature representations are generated, beginning with the input pictures at the bottom (zeroth) level, $f_t^0 = P_t$. Convolutional filters are used to downsample the features from the previous $l-1th$ pyramid level, f_t^{l-1}, in order to create the feature representation at the previous lth layer, f_t^l by a ratio of 2. The feature channels in the first through sixth layers of the pyramid are 16, 32, 64, 96, 128 and 196, in that order.

Warping Net The features of the second picture towards the first image are warped at the lth level using the $\times 2$ upsampled flow from the $l+1th$ level.

$$k_w^l(y) = k_2^l(y + up_2(w^{l+1})(y)) \quad (1)$$

where the upsampled flow, $up_2(w^{l+1})$ is initialized to zero at the highest level and y is the pixel index. As outlined in [34], the gradients to the input CNN features and flow for backpropagation are computed to carry out the warping operation using bilinear interpolation. Warping may be used to verify that picture patches are scaled accurately and to reduce certain geometric distortions in motion that includes transformations other than translation.

Cost Volume Net The features are used to create a cost volume, matching expenses for connecting every pixel in the next frame with its associated pixels [20]. According to [15, 55], the connection between the characteristics of the first picture and the distorted features of the second image gives the definition of the matching cost.

$$CV^l(x_1, x_1) = \frac{1}{N}\left(c_1^l(x_1)\right)^{\top} c_w^l(x_2) \quad (2)$$

The transpose operator is represented by $\top$, while the column vector length $c_1^l(x_1)$ is shown by N. It is only required to compute a partial cost volume within a limited range of d pixels in an $L-$level pyramid setup. For example, $|x_1 - x_2|_\infty \leq d$. In the full-resolution photos, a single-pixel motion at the highest level corresponds to $2L-1$ pixels. As a result, we may set d to a low number. The 3D cost volume has dimensions of $d^2 \times H^l \times W^l$, where H^l and W^l stand for the lth pyramid level's height and breadth, respectively.

Estimator Net for Optical flow Inputs to this multi-layer CNN include the cost volume, first-image features, and upsampled optical flow; outputs include the flow W^l at the lth level. With values of 128, 128, 96, 64, and 32 for each convolutional layer, the number of feature channels is constant throughout all pyramid levels. Instead of sharing common parameters, estimators at different levels have their own set of them. Iteration of this estimating procedure continues until the target level, l_0.

B. Feed forward Context Network

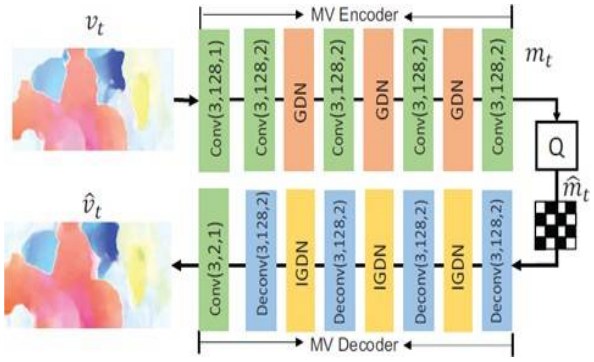

Figure 3. Structure of feed forward context network.

Our method involves using a feedforward CNN-based Con-text network, first presented in the [41], to compress pixel-leveloptical flow v_t from the motion estimation network for picture compression. Using optical flow v_t, the

context network creates the motion representation m_t. A series of nonlinear transformations and convolution operations are applied to the optical flow v_t. With the exception of the final deconvolution layer, which has a size of 2, all layers have 128 output channelsfor convolution (deconvolution). Quantizing m_t results in $\hat{m}_t$ from the motion representation. For entropy coding, the quantized representation $\hat{m}_t$ is used. The network of context encoders and decoders makes it possible to compress optical flow effectively. Figure 3 shows the feed forward compressionnetwork. Conv(3,128,2) is the symbol for the 3x3 kernel convolution operation, 128 output channels, and a stride of 2. GDN/IGDN refers to the nonlinear transform function.

C. The PWC Net Algorithm

The algorithm for optical flow estimation process usingthe PWC-Net is provided below. This simplified algorithm provides an overview of the key steps involved in PWC-Net's optical flow estimation process that includes building feature pyramids, computing cost volumes, estimating flow using CNNs, warping and utilizing context networks for refinement.

Algorithm 1 To find optical flow between the input frames

Input: Two input frames, I_1 and I_2
Output: Estimated optical flow between I_1 and I_2
procedure FINDOPTICALFLOW(I_1, I_2)
    BuildFeaturePyramids(I_1, I_2)
    for each pyramid level (starting from the top) **do**
        ComputeCostVolume($I_{1(features)}$, $I_{2(features)}$)
        EstimatedFlow ← Estimate Optical Flow(CostVolume, $I_{1(features)}$)
        UpsampleAndRescale(EstimatedFlow)
    end for
    for the second-to-top pyramid level **do**
        WrapFeatures($I_{2(features)}$, EstimatedFlow)
        ComputeCostVolume($I_{1(features)}$, Wrapped $I_{2(features)}$)
        EstimatedFlow ← EstimateOpti-calFlow(CostVolume, $I_{1(features)}$, EstimatedFlow)
    end for
    for each lower pyramid level **do**
        WrapFeatures($I_{2(features)}$, EstimatedFlow)
        ComputeCostVolume($I_{1(features)}$, Wrapped $I_{2(features)}$)
        EstimatedFlow ← EstimateOpti-calFlow(CostVolume, $I_{1(features)}$, EstimatedFlow)
    end for
    RefineFlow(EstimatedFlow)
    Output EstimatedFlow
end procedure

IV. EXPERIMENTS AND RESULTS

This section gives the detailed description of the proposed framework for the motion estimation phase in video compression. In this segment we explain the PWC-network to generateoptical flow and PWC-Net's fundamental components, Contextnetwork to post process and compress the generated optical flow.

The proposed PWC-Net model is developed using Python programming language. A Python function named *Motion_Estimation*, performs motion estimation using the PWC-Net optical flow model. The function takes two parameters: *rep_fr*, representing the number of frames to process, and *MODEL_dst*, representing the destination path for saving the flow and frame data, working of which is as follows:

1. If the TRAIN_MODE variable is set to *True*, the function enters the training mode and proceeds to collect information and training data. It reads a video file frame by frame, calculates the optical flow using the PWC-Net model, and saves the resulting flow images and frames to the specified destination path. In the training mode, the function iterates through each frame of the video, generates predictions using the PWC-Net model, and saves the optical flow images and frames to the destination path. It also saves the flow data and frame data as NumPy arrays.
2. If the TRAIN_MODE variable is set to *False*, the functionenters the inference mode. In this mode, it loads pre-trained flow and frame data from the specified destination path and displays the raw images and flow images using OpenCV.
3. The function returns the flow data and frame data.

The proposed method in general performs motion estimationusing the PWC-Net optical flow model, allowing for bothtraining and inference modes and enabling the processing of video data for various applications such as video compression or motion analysis.

TABLE I. SIMULATION DATA OF THE DATASET VIDEOS

Dataset Category	Video format	Size	Duration /Length	Resolution	Frame rate (frames/sec)	No. of frames considered	Bitrate (kbps)	Data rate (kbps)
Self-recorded reading video	MPEG-4	18 MB	10s	1920 × 1080	23.90	960299	1379	13550
News video 1	MPEG-4	2878 KB	10s	1280 × 720	25	249	2349	2089
News Video 2	MPEG-4	568 KB	5s	1280 × 720	25	125	922	791
News video 3	MPEG-4	432 KB	4s	1280 × 720	25	112	699	583

*Training Datasets***.** The PWC network was trained on MPEG-4 recorded videos and few news videos sourced from YouTube, each with a duration of approximately 5 to 10 seconds with varying resolutions and frame rates to generate the optical flow, the details of which are tabulated in table 1.For training, a random 128x128 patch consisting of 49 frames was selected from each video, and the input video values were normalized to [-1,1]. To minimize any existing compression artifacts, the original frame was randomly down sampled, anda 128x128 patch was extracted.

Results. After training the dataset videos, each video is fed into the proposed PWC-Network model to generate the opticalflow that is considered as motion vector. The Figure 4 depicts the raw image and the flow image generated by the proposed model for the datasets considered as mentioned in the table 1.

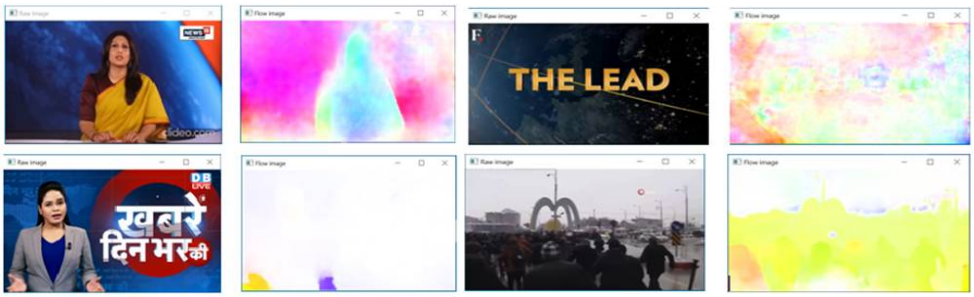

Figure 4. Left: Raw image (original image) Right: Flow image generated by the proposed PWC-Net model for the sample datasets.

V. CONCLUSION AND FURURE WORK

Various algorithms and methods are used for motion estimation, ranging from traditional block-based approaches to more advanced techniques based on optical flow and machine learning. In our work, we have created a concise yet highly effective CNN model called optimized PWC-Net for motion estimation through optical flow that relies on fundamentalprinciples: pyramidal processing, warping and the utilization of cost volume. By combining deep learning with domain- specific knowledge, we have not only reduced the model'ssize but also enhanced its performance. Given its compactness,efficiency and effectiveness, PWC-Net is poised to be a valuable component in numerous video processing systems. There is always a scope for further enhancements and innovationsas innovation is an ongoing process that will persist and accelerate as scientific and technological fields advance to the forefront of cutting -edge developments. Ultimately, the goalis to find the optimal trade-off between compression efficiencyand computational complexity to ensure high-quality video compression. As a part of future work, implementation will be carried out on next stages of video compression namely, Motion compensation to predict new frame (pattern image), followed by transformation and quantization step to get the quantized image and finally, entropy coding to encode the quantized information obtained from quantization phase to get a compressed video.

REFERENCES

[1] Chen M, Goodall T, Patney A, Bovik AC. "Learning to compress videos without computing motion", Signal Processing: Image Communication, 2022 Apr 1;103:116633.

[2] D. Liu, Y. Li, J. Lin, H. Li, and F. Wu, "Deep learning based video coding: A review and a case study," arXiv preprint arXiv:1904.12462, 2019. 1, 2.

[3] C.-Y. Wu, N. Singhal, and P. Krahenbuhl, "Video compression through image interpolation," in Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 416–431.

[4] T. Chen, H. Liu, Q. Shen, T. Yue, X. Cao, and Z. Ma, “Deepcoder: A deep neural network based video com- pression”, IEEE Visual Communications and Image Processing (VCIP). IEEE, 2017, pp. 1–4.
[5] Guo Lu, Wanli Ouyang, Dong Xu, Xiaoyun Zhang, Chunlei Cai, and Zhiyong Gao, “DVC: An End-to-end Deep Video Compression Framework”, IEEE Transactions on Pattern Analysis and Machine Intelligence, April 2020
[6] R. Yang, F. Mentzer, L. Van Gool, and R. Timofte, “Learning for video compression with hierarchical quality and recurrent enhancement,” arXiv preprint arXiv:2003.01966, 2020.
[7] 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 conference on computer vision and pattern recognition 2018 (pp. 8934-8943), preprint arXiv:1709.02371, 2017. preprint, original paper is published on CVPR, June 2018.
[8] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski, “A database and evaluation methodology for optical flow”, International Journal of Computer Vision (IJCV), 2011. 1, 2, 3
[9] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, “A naturalistic open-source movie for optical flow evaluation”, In European Conference on Computer Vision (ECCV), 2012. 1, 3
[10] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? The KITTI vision benchmark suite”, In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012. 1, 3
[11] B. Horn and B. Schunck, “Determining optical flow”, Artificial Intelligence, 1981. 1, 2, 3, 4.
[12] Dosovitskiy A, Fischer P, Ilg E, Hausser P, Hazirbas C, Golkov V, Van Der Smagt P, Cremers D, Brox T, “Flownet: Learning optical flow with convolutional networks”, In Proceedings of the IEEE international conference on computer vision 2015 (pp. 2758-2766).
[13] E. Memin and P. Perez, “Dense estimation and object- based segmentation of the optical flow with robust techniques”. 7(5):703–719, May 1998. 2
[14] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, “High accuracy optical flow estimation based on a theory for warping”, In ECCV, Springer, 2004. 2 volume 3024, pages 25–36.
[15] A. Wedel, D. Cremers, T. Pock, and H. Bischof, “Structure and motion-adaptive regularization for high accuracy optic flow”, In ICCV, Kyoto, Japan, 2009. 2
[16] T. Brox and J. Malik, “Large displacement optical flow: descriptor matching in variational motion estimation”, PAMI, 33(3):500–513, 2011. 2, 4, 6, 7
[17] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid, “DeepFlow: Large displacement optical flow with deep matching”, In ICCV, Sydney, Australia, Dec. 2013. 2, 7
[18] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid, “EpicFlow: Edge-Preserving Interpolation of Correspondences for Optical Flow”, In CVPR, Boston, United States, June 2015. 2, 7
[19] D. Rosenbaum, D. Zoran, and Y. Weiss, “Learning the local statistics of optical flow”, In NIPS, 2013. 2
[20] D. Sun, S. Roth, J. P. Lewis, and M. J. Black, “Learning optical flow,” in European Conference on Computer Vision (ECCV), 2008. 3
[21] M. J. Black, Y. Yacoob, A. D. Jepson, and D. J. Fleet, “Learning parameterized models of image motion”, In CVPR, 1997. 2
[22] M. Leordeanu, A. Zanfir, and C. Sminchisescu, “Locally affine sparse-to-dense matching for motion and occlusion estimation”, ICCV, 0:1721–1728, 2013. 2
[23] R. Kennedy and C. Taylor, “Optical flow with geometric occlusion estimation and fusion of multiple frames”, In EMM-CVPR. 2015. 2
[24] K. R. Konda and R. Memisevic, “Unsupervised learning of depth and motion”, CoRR, abs/1312.3429, 2013. 2
[25] G.W. Taylor, R. Fergus, Y. LeCun, and C. Bregler, “Convolutional learning of spatio-temporal features”, In ECCV, pages 140–153, 2010. 2
[26] E. P. Simoncelli, E. H. Adelson, and D. J. Heeger, “Probability distributions of optical flow,” in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1991. 3
[27] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael, “Learning low-level vision”, International Journal of Computer Vision (IJCV), 2000. 3
[28] S. Roth and M. J. Black, “On the spatial statistics of optical flow,” International Journal of Computer Vision (IJCV), 2007. 3
[29] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski, “A database and evaluation methodology for optical flow”, International Journal of Computer Vision (IJCV), 2011. 1, 3, 4, 8.
[30] Y. Li and D. P. Huttenlocher, “Learning for optical flow using stochastic optimization,” in European Conference on Computer Vision (ECCV), 2008. 3
[31] M. J. Black and P. Anandan, “The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields”, Computer Vision and Image Understanding (CVIU), 1996. 2, 3, 4, 5
[32] J. Wulff and M. J. Black, “Efficient sparse-to-dense optical flow estimation using a learned basis and layers,” in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 120–130. 3
[33] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox, “A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation,” in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016. 3

[34] 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 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017. 1, 2, 3, 4, 6, 8, 10, 11
[35] A. Ranjan and M. J. Black, “Optical flow estimation using a spatial pyramid network,” in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017. 1, 3, 8, 11
[36] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, “Extracting and composing robust features with denoising autoencoders”, In International Conference on Machine Learning (ICML), 2008. 3
[37] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation”, In International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI), 2015. 1, 2, 3
[38] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “DeepLab: Semantic image segmentation with deep convolutional netsatrous convolution, and fully connected CRFs”, IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2017. 3
[39] G. Huang, Z. Liu, K. Q. Weinberger, and L. vander Maaten, “Densely connected convolutional networks”, In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017. 3, 4
[40] S. Jegou, M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio, “The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation”, In IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshop, 2017. 3
[41] J. Ball e, V. Laparra, and E. P. Simoncelli, “End- to-end optimized image compression,” in 5th International Conference on Learning Representations, ICLR, 2017. 1, 2, 5, 6, 15, 16.