Lightweight Network For Video Motion Magnification

Jasdeep Singh, Subrahmanyam Murala and G. Sankara Raju Kosuru
Computer Vision and Pattern Recognition Lab
Indian Institute of Technology Ropar, India
{jasdeep.19eez, subbumurala, raju} @iitrpr.ac.in

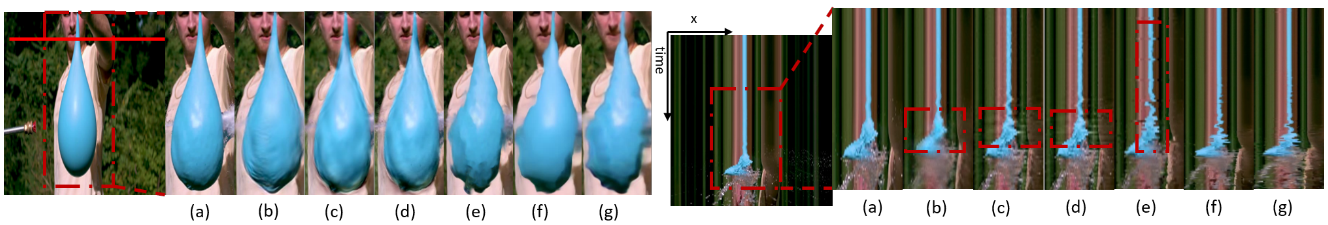

Figure 1. Balloon video: First frames from the video are shown and next to them temporal slices taken from the red strip are illustrated for visualization of balloon bursts motion. Motion magnification can be perceived as more motion in the balloon (also visible in the temporal slice) as compared to the input. While the other methods produce distortions such as ringing artifacts, spurious motion *etc* (highlighted in the red box). The proposed method produces better magnification with lesser distortions. (a) Input video, (b) Acceleration based method [39], (c) Jerk-Aware method [30], (d) Anisotropy method [28], (e) Oh *et al.* [23], (f) Ours Base model, and (g) Ours lightweight model. *Please zoom in for a clearer view.* *https://github.com/jasdeep-singh-007/LightweightNetworkForVideoMotionMagnification*

Abstract

Video motion magnification provides information to understand the subtle changes present in objects for applications like industrial, healthcare, sports, etc. Most state-of-the-art (SOTA) methods use hand-crafted bandpass filters, which require prior information for the motion magnification, produces ringing artifacts, and small magnification etc. While others use deep-learning based techniques for higher magnification, but their output suffers from artificially induced motion, distortions, blurriness, etc. Further, SOTA methods are computationally complex, which makes them less suitable for real-time applications. To address these problems, we proposed deep learning based simple yet effective solution for motion magnification. The proposed method uses a feature sharing and appearance encoder for better motion magnification with fewer distortions, artifacts etc. Additionally, for reducing magnification of noise and other unwanted changes, proxy-model based training is proposed. A computationally lightweight model ($\sim$ 0.12 M parameters) is proposed along with the base model. The performance of the proposed models is tested qualitatively and quantitatively, with the SOTA methods. Results demonstrate the effectiveness of the proposed lightweight and base model over the existing SOTA methods.

1. Introduction

The understanding of subtle motion present in dynamic or still objects, is a very challenging task. For example, slight skin deformation occurs while throwing something, small chest movements while breathing, small distortions that occur in objects while moving, *etc*. These small meaningful motion are difficult to see with the naked eye. E.g. as shown in Figure 1, subtle motions generated in balloon while bursting, are hard to perceived with the naked eye, but easy to see in the magnified frames. Due to this, magnification of these changes in the video, become important and result in many industrial and healthcare applications [25], [17], [5], [3], [2], [26], [21], [8]. But these videos also contain noise which is introduced during the photographic process (low light levels, high sensor gain, short exposure time, and so on) [28] . As this noise is at the same level as minute changes, which makes it difficult to distinguish between signal of interest from noise and makes the motion magnification task more challenging.

To address the problem of motion magnification initially, hand-design based approaches were introduced. Many SOTA hand-crafted methods were based on temporal filters which gave good results [36], [33], [34] on static scenarios but they cannot work in dynamic scenarios. To mitigate this, later [39], [30] methods were proposed which can work in both static and dynamic scenarios. But their outputs were prone to ringing artifacts or small magnification *etc*. Also, their filters were not optimal [23]. To solve these issues of hand-crafted filters, the deep learning-based method [23] was proposed. Even without temporal filters, it shows some robustness to noise and produces higher magnification without ringing artifacts. But it has some limitations.

- They extract motion information from shape information to make the network robust to intensity changes. But, their separation of shape information from texture, is not efficient. Sometimes it results in distorted intermediate features which produce unwanted flickering or superious motion.

- Their texture features sometimes deviate much from input textures and this might be responsible for blurry distortions in some frames.

- They did not take computational complexity into account. As real-time applications like respiration rate monitoring, or in industries where time-constrained output is needed, require low latency.

Currently deep learning based approaches in different tasks like deraining, deblurring, object detection [38],[14], [15] *etc* show promise for real-time applications. Inspire by this we propose a lightweight network for video motion magnification. Our proposed lightweight method does not produce unwanted distortions like [23] and is sensitive toward subtle motions. It produces more magnification than SOTA methods in both static and dynamic scenarios. It has a simple yet efficient architecture. Further, different experiments are done to show the qualitative, quantitative analysis, and physical accuracy of the proposed method in comparison to SOTA methods. The main contributions of the proposed work are as follows:

- A lightweight deep learning model is proposed for video motion magnification.

- A feature sharing encoder module is proposed for motion magnification. This module is responsible for appropriate feature map generations for motion extraction and for reducing the effect of the noise before magnification.

- An appearance encoder is proposed to extract common appearance across the frames with its output being restricted by input frames. This module is responsible for appropriate texture synthesis of the output.

- A proxy model based regularization loss is proposed to reduce the magnification of noise and other unwanted changes in motion features.

In the next Section 2, related work to motion magnification is discussed. Further, in Section 3, the proposed method is explained in detail. In Section 4, qualitative and quantitative comparison of natural and synthetic videos is provided.

2. Related Work

Initially, two different approaches were proposed: 1) Eulerian-based motion magnification and 2) Lagrangian-based motion magnification. The Eulerian [36] was a filtering based and Lagrangian, [19] an optical flow-based motion magnification approach. Liu *et al.* [19] suggests the use of Lagrangian based method for video motion magnification for the first time. They assume that in videos, changes that occur in certain object locations over time can be estimated using the optical flow. It extracts the features from the frames and traces those features to cluster them into a group of points, where the changes are magnified. But computing optical flow in this task is expensive. Flotho *et al.* [9] suggest local Lagrangian based motion magnification approach, which was specifically targeted for micro-expression magnification.

Unlike Lagrangian approaches, Eulerian based methods [36], [33], [34], [39], [30] do not explicitly need tracking of object to detect color and subtle motion changes over a fixed point. To magnify color changes Eulerian based methods [36], [39], [30] first decompose the input frames using spatial pyramids. They used gaussian pyramids for color magnification and [33], [34], [39], [30] use complex-steerable pyramids [10] for subtle motion magnification. After spatial decomposition, they apply temporal filter across each pixel at every pyramid level. These bandpass temporal filters help to select the frequency which needs to be magnified and ignore the noise. They generate good magnification results in static scenarios. But, they cannot differentiate between static motion and dynamic changes that occur in the videos. So, they generate distorted, blur output in dynamic scenarios. Recently, different methods were developed to solve this problem [39], [30]. They ignore the large motion and magnify only small variations. However, they have small magnification for subtle changes and depend on narrow band filters for mitigating the effects of noise.

For magnification of meaningful subtle signals, Elgharib *et al.* [7], Verma *et al.* [31], Kooij *et al.* [18] suggest methods that require user intervention or a specific environment. While other methods are independent of these constraints. Verma *et al.* [32] applied the local Laplacian filter (LLP) [24] for better spatial decomposition and to reduce the noise and artifacts. Wu *et al.* [37] used PCA to decompose the input frames and then select the component which best matches spatial variation with the subtle signal that needs to be magnified. But, it requires meaningful changes to be larger as compared to the other changes in a principal component. Takeda *et al.* [28] suggest the use of Fractional

Anisotropy (FA) to magnify meaningful subtle motions and ignore non-meaningful ones. Takeda *et al.* [29] proposed a more accurate temporal filtering while ignoring the large changes as compared to the previous methods. But, all these methods require fine-tuning of hyperparameters from video to video basis. Also, they have small magnification and they did not take occlusion into account [23] *etc*.

To solve the problems related to hand-crafted filters, recently deep learning based approaches were proposed [23], [4], [6]. Chen *et al.* [4] uses gradient ascent to magnify subtle color and motion changes, but it has small magnification and requires a lot of pre-processing. Nowara *et al.* [22] use [4] and explore the possibility of motion magnification as a pre-processing task in recovering the photoplethysmogram. Dorkenwald *et al.* [6], disentangle shape and appearance features. But, generating output on different scenarios, it requires training on videos of that respective scenario [6]. Oh *et al.* [23] proposed the use of synthetic data to train a deep neural network. It takes two frames and a magnification factor as an input at a time to produce a motion magnified output frame. It gives better noise performance and more magnification as compared to other methods [23] by using only two frames. However, sometimes it produces spurious motion. Also, these methods are computationally complex, which makes it difficult to use them in different real-time healthcare or industrial applications.

3. Proposed Method

In the subsequent subsections first, the proposed method is explained in detail. Later, the final loss function, training dataset, procedure, and the proposed base and lightweight model are discussed.

3.1. Network Architecture

We propose a lightweight deep learning based network to magnify the subtle motions in the videos. It consists of encoder-decoder based architecture. It uses two feature sharing based encoders, to translate input frames from image space to feature space where motion information can be extracted. Handcrafted methods [33], [34], [39], [30] use complex steerable pyramids for the same task. But, Oh *et al.* [23] uses simple encoders and gives its features to shape encoders to extract shape features. It extracts motion information from the shape features. For separating shape information from image features, it puts regularization across the encoders to constrain the feature space. Instead of that, we let the network decide the encoding feature space for motion extraction.

A major issue with motion magnification is to reduce the effects of changes due to noise, illumination *etc* while magnifying meaningful changes. This is a hard problem. Hand-crafted methods [33], [34], [39], [30] depend on narrow band pass filter (which require prior information about the frequency of interest). Whereas Oh *et al.* [23] method presumes that noise, unwanted illumination *etc* changes are part of intensity changes and motion information is present in shape changes. So, they try to separate shape from texture representation (intensity information). For this, while training the network they provide intensity perturbed frames that have the same shape information as un-perturbed frames. Then they take L_1 loss across perturbed and un-perturbed frames features. They assume that shape information across intensity change should remain the same. They take the difference between these shape features, magnify it and add it to the texture encoder features. But their method is not efficient. It sometimes results in distorted intermediate features which produce flickering or superious motion. Whereas the proposed method uses feature sharing encoder for the motion extraction and proxy model based feature loss with appearance encoder loss to reduce the effects of noise before magnification. The denoising signal in network training, comes from three different places 1) from the final predicted output, 2) common appearance based regularization loss 3) proxy model based feature loss. Jointly optimizing across these losses helps to reduce the effects of noise in motion magnification (a detailed discussion is given in section below). The manipulator multiplies the motion features to the magnification factor (which decides the amount of magnification), and apply non-linear transforms using residual blocks. The manipulator output is added to the common appearance encoder output and given to the decoder. The decoder converts intermediate features to image space and generates the final magnified output. Figure 2 (A) describes the proposed model.

Feature Sharing Encoder ($E(.)$): Feature Sharing Encoder is used to reduce the effect of noise before magnification (decoder is used to reduce the effect of noise after magnification). We assume different frames will have distinct noise. With concatenation operation across features, each encoder will have information about the input frames and improved features of the other encoder. The network can compute weighted averages to decrease the effects of illumination, noise *etc*. It's also used to convert the input from image space to feature space for motion extraction. Unlike [23], its output features (E_a, E_b) are not restricted by regularization. Residual blocks [11] are used to map input frames to a feature space where motion information is extracted by taking the feature differences as shown in Figure 3. Max-pooling is used to down-sample the features to reduce the computation and increase the receptive field. The feature sharing encoder is illustrated in Figure 2 (A).

Appearance Encoder ($A(.)$): Relevant texture content is required to combine with motion information to generate the magnified frame. For generating texture content, [23]

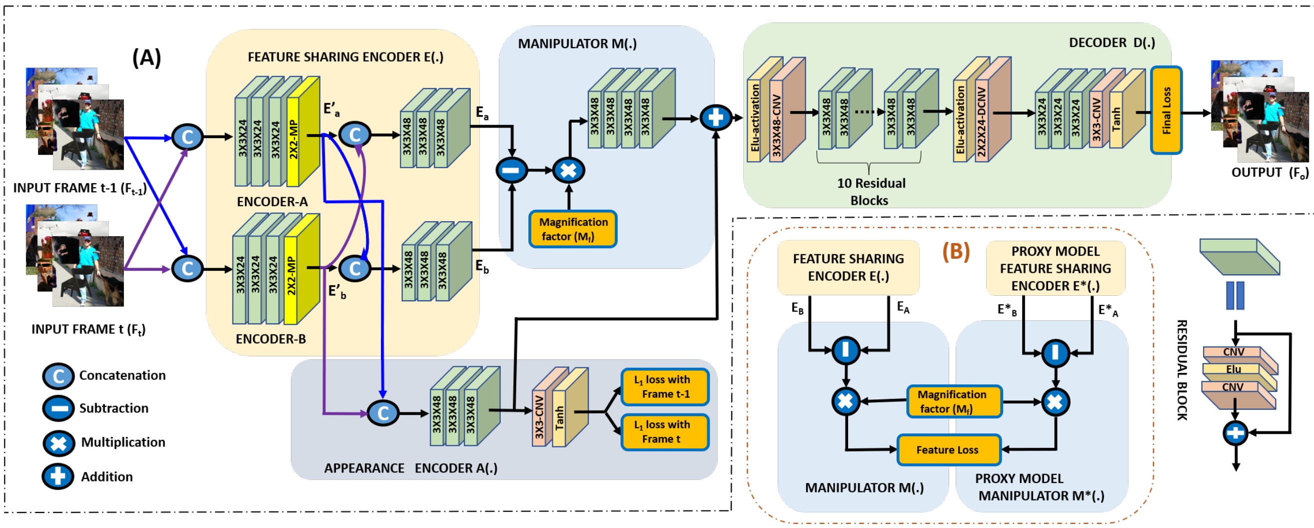

Figure 2. **(A)** Proposed deep learning model for motion magnification. It consists of feature sharing encoder, appearance encoder, manipulator, and decoder. F_t and F_{t-1} the two consecutive frames, with M_f as the magnification factor, are given as input to the network. F_o is the magnified output frame. Residual Blocks with 3×3×48 show that there is a 3 × 3 convolution filter with 48 channels, similarly for Residual Blocks with 3×3×24. E_a and E_b are the output features and E'_a and E'_b are the intermediate features of the feature sharing encoder. **(B)** Proxy model feature loss across the manipulator block. *Please zoom in for a clearer view.*

proposes a regularization term to minimize the difference in texture feature representation between the frames. To satisfy this regularization term both texture encoders with different input tries to generate a common representation. But this representation can deviate from actual texture representation. We assume this can be the probable reason for producing texture distortion (blurry distortions) sometimes. To solve this, we propose Appearance Encoder ($A(.)$). Generally, the magnified frame has a high correlation with the input frames as most of the objects are still. In $A(.)$ we exploited this fact for appropriate texture generation. Loss between appearance encoder $A(.)$ features and input frames are used to extract common appearance features. This also, prevents the learn able parameters to generate features that deviate from F_t and F_{t-1}. For calculating this loss, no noise is added to the ground truth (input frames). So, it will also force denoising characteristics in common texture features. This will help in the better generation of the output. Both encoder intermediate features E'_a and E'_b (as shown in Figure 2 (A), as the output of both encoders) are concatenated (ζ represents the concatenation operation) and is given as input to the appearance encoder. Then residual blocks are applied on them for feature transformation to produce output $A(\zeta(E'_a, E'_b))$. The regularization loss L_A between input frames F_t, F_{t-1} and appearance encoder output $A(\zeta(E'_a, E'_b))$ is defined in Eq. (1)

$$L_A = |\phi(A(\zeta(E'_a, E'_b))) - F_t)|_1 + \\ |\phi(A(\zeta(E'_a, E'_b))) - F_{t-1})|_1 \tag{1}$$

where ϕ represents the convolution operation with $3 \times 3 \times 3$ filters and tanh activation.

Manipulator ($M(.)$): We assume motion information can be extracted from the difference in encoder features. This is somewhat different from [23] assumption, where they presume motion information can be extracted from the difference of encoder shape features. The manipulator (M) gets the non-linear transformed encoder shared features of E_a and E_b as input. It takes their difference and multiplies them with the magnification factor M_f. Then these features are given to residual blocks for non-linear transformations to generate output $M((E_a - E_b) \times M_f)$ (the structure of manipulator is similar to [23]). Figure 3 shows the difference features of the feature sharing encoder block that highlight the motion information.

Decoder: The combined output of the appearance encoder and manipulator is given to the decoder as shown in Figure 3. In the decoder, ten residual blocks before up-sampling are used, as they decrease the computation requirements and increase the receptive fields. The up-sampled features are passed through three residual blocks. In the end, a convolution layer with 3×3 filter size and tanh activation is used to generate the magnified output F_o (the structure of the decoder is similar to [23]).

Proxy Model Based Feature Loss: The proxy model has the same architecture as the proposed model but it is trained

Figure 3. (a) depicts the input frames, (b) shows the motion features (after subtraction of encoder features). These features highlight the object of motion.

without adding noise. The proxy model features (of the noiseless image) are taken as the ideal features and the main model features (with noisy input) should try to get close to it (It can also be viewed as teacher-student training paradigm [12], [20], [13], where the teacher has the same network as a student, but teacher network is used to pass denoising information to the student network in the feature space). At the time of training of the main model, proxy model weights are fixed. For calculating the distance between proxy model noiseless features and main model noisy features L_1 loss is used. This feature space loss is only sensitive to noise present in the motion information. Whereas the appearance encoder loss term is sensitive to noise present in texture, and predicted output loss terms are sensitive to the magnified noise (particularly which can cause large variations after magnification). So, proxy model based feature loss will help to make motion information more robust. Loss is taken in between the manipulator features after subtraction and multiplication with magnification factor $((E_a - E_b) \times M_f)$ as shown in Figure 2 (B). We assume that this will help to prevent any distortions that can be generated due to magnification of noise, illumination changes *etc*. Proxy model based feature loss can be defined as follows:

$$L_M = |((E_a^* - E_b^*) \times M_f) - ((E_a - E_b) \times M_f)|_1 \quad (2)$$

where superscript notation $*$, indicates the proxy model.

Final Loss Function: We consider the L_1 loss, loss between edges (L_{edge}) and Perceptual Loss (L_p) for bettering of output quality. The L_1 loss computes the pixel level difference of predicted label $\hat{y}$ and ground truth y. L_1 loss is illustrated as

$$L_1 = \sum |\hat{y} - y|_1 \quad (3)$$

In the motion magnification problem, the L_1 loss is less sensitive to object motion because most of the region in output frames does not have motion. Further, there may exist many minima in L_1 which produce blur output [35] around the motion parts (near the edges). So, to put more focus on the edges of the output, we take the loss between the edges of the predicted and ground truth frames (L_{edge}), (as defined in [1]). L_{edge}, helps to make the model more sensitive towards the edges [1] of the reconstructed motion magnified frames. L_{edge} is given as

$$L_{edge} = \sum |\nabla \hat{y} - \nabla y|_1 \quad (4)$$

∇ shows the finite differences in a horizontal and vertical direction [1] for computing edges. Another issue with the texture of the moving object is that there still exist many minima which can give low loss but with bad perceptual quality. For this, a loss in a higher dimension is needed. Hence, to increase the perceptual quality of the motion magnified frames, we use the perceptual loss (L_p) [16] along with the L_1 and L_{edge}. The L_p is given as

$$L_p = \sum |\phi_i(\widehat{y}) - \phi_i(y)|_1 \quad (5)$$

Where, ϕ_i represents the VGG-16 [27] feature space activations. The final loss of the proposed network (L_{total}) is given in Eq. (6)

$$L_{total} = \lambda_1 L_1 + \lambda_2 L_p + L_{edge} + L_A + L_M \quad (6)$$

Where λ_1 and λ_2 are the weights for L_1 loss and Perceptual Loss (L_p) [16] respectively. $\lambda_1 = 10.0$, and $\lambda_2 = 0.1$ values are considered for the network training and they are determined experimentally.

Dataset and Training: The proposed models, base model, and lightweight model are trained on the training dataset provided by [23]. In the network, C channels are used in primarily layers, and after down-sampling $C \times 2$ channels. For base model $C = 24$ and for lightweight

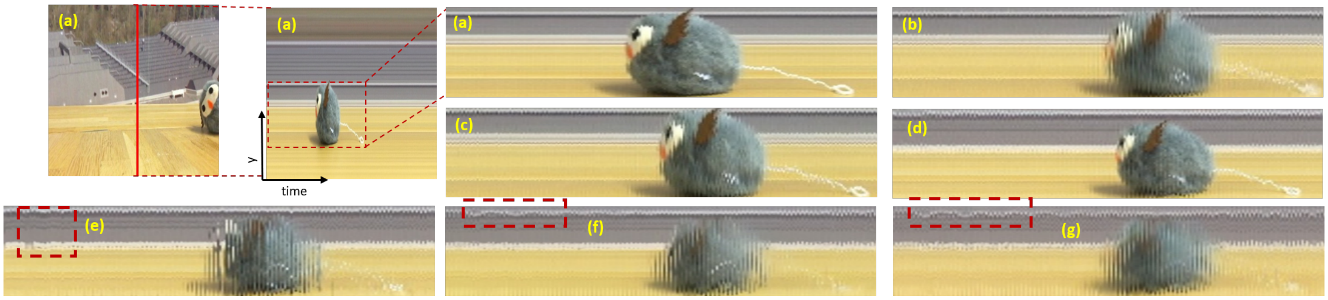

Figure 4. A toy is vibrating and moving along the table from right to left. The spatial-temporal slices from the respective methods are taken from the red strip. The proposed method shows more magnification (also higher motion of the background is highlighted in the red bounding boxes). (a) Input video, (b) Acceleration based method [39], (c) Jerk-Aware method [30], (d) Anisotropy method [28], (e) Oh *et al.* [23], (f) Ours Base model, and (g) Ours lightweight model.

Table 1. Comparison of the SOTA learning method [23] with the proposed base network (M_1) and the lightweight network (M_2) in terms of number of parameters, FLOPS, and run time. (Run time values are calculated at 720X720 resolution on NVIDIA 2080 RTX for higher quality output).

Model	**Parameters**	**GFLOPs**	**Run Time**
Oh *et al.* [23]	0.98M	268.6	95 ms
M_1	1.10M	375.5	142 ms
M_2	0.12 M	42.4	38 ms

model $C = 8$ is considered. For training, the learning rate is set to .0001, and an ADAM optimizer is used. Models are trained for 47 epochs. The proposed lightweight model has 7.6 $\times$ lesser parameters and 6.3 $\times$ lesser flops as compared to [23] as show in Table 1.

4. Experimental Results

The proposed model is evaluated qualitatively and quantitatively on real-life and synthetic videos and is compared with the SOTA methods [30], [23], [39], [28] for motion magnification (linear filter based method [33] is not considered for comparisons as they produce distortions in dynamic scenarios). Also, an ablation study is conducted to show different aspects of the proposed method. With least computational complexity, the proposed lightweight model provides better results than SOTA methods. The detailed discussion is given in the following subsections.

4.1. Analysis on Real Videos

Analysis on Balloon video: In the balloon video, a water cannon is fired on a balloon to rupture it, as shown in Figure 1. Due to this, small motions are developed in the balloon along with its large bursting motion. Our aim is to magnify the minute balloon motion while producing minimum distortions due to sudden large motion. Figure 1 shows the motion of the balloon at the red strip along time. Hand-crafted methods [30, 28, 39] create ringing artifacts along the balloon (*visible as white edges near the balloon and white spikes in the temporal slices highlighted in the red boxes, in Figure 1*). Further, Oh *et al.* [23] produce blurry distortions in some frames (in the balloon and the background object), visible as spikes in the temporal slice (*illustrated in red bounding box in Figure 1 temporal slice).* Whereas, the proposed method shows better magnification with lesser distortions around the balloon.

Analysis on Toy Video: The toy video is illustrated in Figure 4. In this video, the toy is moving on the table along with vibrations. Our goal is to produce large magnification for the toy's subtle motions in presence of toy linear motion (moving along the table from left to right). The Jerk-aware [30], Acceleration [39] and Anisotropy [28] methods produce less magnification. Further, the Acceleration [39] and Oh *et al.* [23] produce some blurriness in the output. Oh *et al.* [23] method produces good magnification but causes spurious motion *(visible in red box as sharp spikes in Figure 4 (e)).* Whereas, our proposed models produce better magnification of the vibrating toy as compared to [30], [39], [28], [23].

Analysis on Gun-shooting Video: Figure 6 show the results of different SOTA methods on gun-shooting video. This video contains a large background movement due to camera motion and quick gun recoil produces the foreground motion. Our aim is to magnify the minute forearm motion in presence of a large camera motion. Figure 6 shows the motion of the forearm using spatio-temporal slices at a red strip. Higher forearm motion can be perceived as more bending in the temporal slice (shown in the red box of Figure 6). Jerk -aware method [30], Anisotropy [28], Acceleration [39] methods produce lower magnification as compared to the proposed method. Oh *et al.* method [23] induce spurious motion in some frames and generate blurry distortions (*visible as large spikes in Oh et al. [23] temporal slice).* Whereas, the proposed method generates higher magnification of subtle forearm movements with fewer dis-

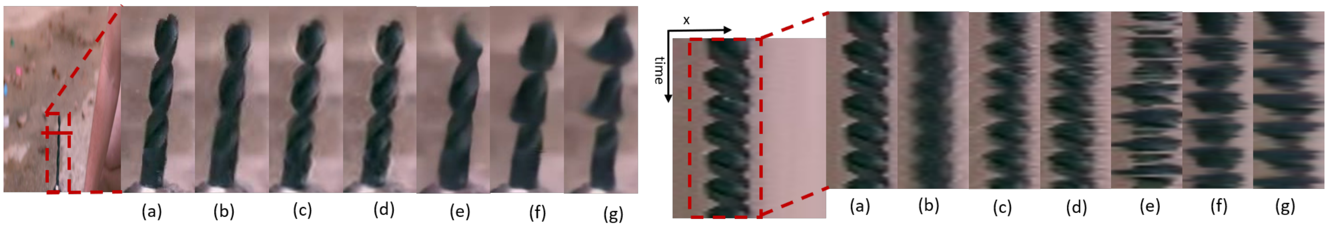

Figure 5. Drill Video: Comparison of proposed method with existing methods for magnification of the drill rotational motion. First, output from respective methods and then their spatio-temporal slices with respect to the red strip are shown. The proposed method produces better results with fewer artifacts. (a) Input video, (b) Acceleration based method [39], (c) Jerk-Aware method [30], (d) Anisotropy method [28], (e) Oh *et al.* [23], (f) Ours Base model, and (g) Ours lightweight model.

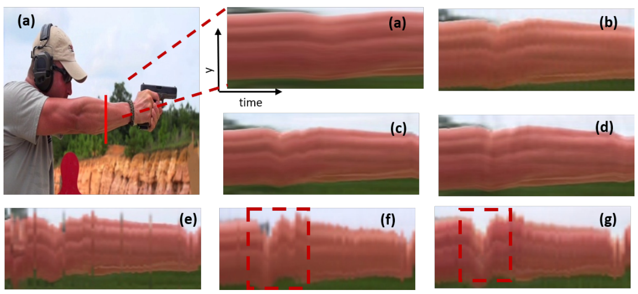

Figure 6. Gun-shooting video: Visualizing the impact of gun recoil through the arm. We take temporal slices at red strip to show the effect of magnification on the forearm. The proposed method output has the highest magnification (shown as more bending of the forearm in the red box). (a) Input video, (b) Acceleration based method [39], (c) Jerk-Aware method [30], (d) Anisotropy method [28], (e) Oh *et al.* [23], (f) Ours Base model, and (g) Ours lightweight model.

tortions, even in presence of large camera motion as compared to SOTA methods.

Analysis on rotational motion: Figure 5 illustrates a hand drill producing rotational motion along its axis. To analyze the effects of magnification on rotational motion a still video is taken. In 2D, hand drill rotational motion can be perceived as spiral motion. Our aim is to increase the spiral motion (higher spiral motion is displayed as more outwards extension of rod radius). The rotational motion of the hand drill is depicted in spatial temporal slice of Figure 5. Hand design filter-based methods [30, 28, 39] generate ringing artifacts around the rod *(seen as white edges near the rod and white spikes in the temporal slices in Figure 5 (b),(c),(d))*. Oh *et al.* method [23] magnifies the motion but delivers some distortions in the magnified frames (observable as white spikes in Figure 5(e) temporal slice). Our proposed models have better magnification and fewer artifacts in motion as compared to SOTA methods.

Whether our magnified output is physically accurate? To check the physical accuracy of the proposed method, we

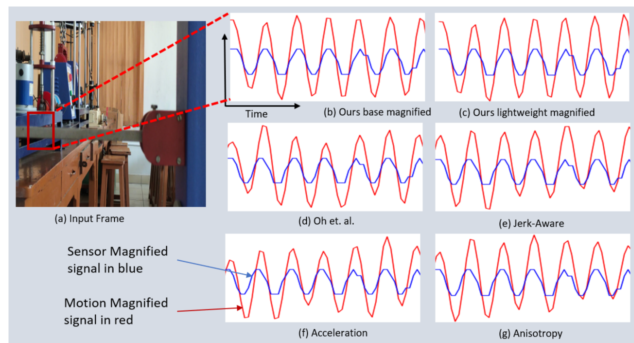

Figure 7. Physical Accuracy: Comparison between our method and other SOTA methods output (in red) with the sensor signal (in blue) respectively. The direction of optical flow in the patch region is computed to extract the magnified signal (in blue) from the video. (a) Input, (b) Our base model (c) Our lightweight model, (d) Oh *et al.* method [23] (e) Jerk-aware method [30], (f) Acceleration method [39] and (g) Anisotropy [28] method respectively.

Table 2. Mean Absolute Error (MAE) on SOTA methods of Anisotropy [28], Jerk-aware method [30], Acceleration method [39], Oh *et al.* method [23], Ours base method (M_1), and Ours lightweight model (M_2). MAE is computed between the extracted signal from the magnified video and sensor measured signal. The proposed method has the minimum MAE values. (First best shown in bold and second best shown in italic.)

Methods	[28]	[30]	[39]	[23]	M_1	M_2
MAE	0.146	0.149	0.146	0.144	**0.121**	*0.131*

perform this experiment. A mechanical rod as shown in Figure 7 is displaced up and down using universal vibration apparatus. An ultrasonic sensor is used, to measure the displacement signal of the mechanical rod and at the same time, it is recorded in the video. For extraction of motion signal from the video first, the optical flow is computed by taking input frame t-1 and magnified frame t along the region marked in the red box in Figure 7. Then the average direction of motion along the image patch is calculated. Both the optical flow and sensor measure signal is rescaled from 0 to 1. From the rescaled signal, mean absolute error

Table 3. Aggregate Mean Square Error (MSE) of synthetic videos with different backgrounds on Acceleration method [39], Jerk-aware method [30], Anisotropy [28], Oh *et al.* method [23] Ours base method (M_1), and Ours lightweight model (M_2) respectively. The proposed method has the minimum error. (First best shown in bold and second best shown in italic.)

Methods	[28]	[30]	[39]	[23]	M_1	M_2
MSE	36.4	55.3	68.0	38.8	**23.07**	*27.8*

Table 4. Aggregate Mean Square Error (MSE) computed across synthetic videos on (a) Without proxy model based feature loss training, (b) Without feature sharing encoder, (c) Without appearance Encoder, (d) Without L_{edge} loss, (e) Without L_p loss and (e) Ours base model (M_1) on synthetic videos. The proposed method has the minimum error. (First best shown in bold.)

Methods	M_1	(a)	(b)	(c)	(d)	(e)
MSE	**23.07**	27.85	30.1	37.7	31.1	40.2

(MAE) is calculated for different SOTA methods as shown in Table 2. The proposed method has the minimum MAE.

4.2. Analysis on Synthetic Videos

For quantitative analysis, we generate 25 different synthetic videos with various backgrounds. To mimic photographic noise, Gaussian noise is also added in the videos. This will help to see how each method behaves in different backgrounds and their robustness towards the noise. Each video contains three circles to mimic motion in a different direction (one in horizontal, one in vertical, and one in diagonal). This will help to analyse how different methods [30], [23], [28] and [39] magnify motions individually. For synthetic videos, we choose a simple range of motion (up-down, left-right, and diagonal motion of circle). As it is easier to generate accurate ground truths for simpler cases, compared to the complex motion (random movement of a circle). Input videos contain a circle with 0.1 sub-pixel movement, while the ground truth has 10 pixels movement. For different methods, to generate the same amount of output motion, their magnification factor is changed. Table 3 depicts the average MSE of 25 different synthetic videos, on different SOTA methods [30], [23], [28], [39] and ours. Our method produces better results with minimum aggregate MSE.

Ablation Study: Ablation study is conducted on the proposed architecture to see the importance of different modules. For this, five different models are trained (a) Without proxy model based feature loss training (b) Without feature Sharing Encoder, (c) Without appearance Encoder, (d) Without L_{edge} loss, and (e) Without L_p loss. We test them on synthetic videos and give their aggregate MSE in Table 4. The proposed method shows the minimum MSE value.

The proposed feature based proxy loss is used to reduce the magnification of unwanted changes. Appearance encoder based loss helps to give denoising signal to make the network robust to illumination changes. Further, feature sharing encoder is used to reduce the effects of noise. Also, the appearance encoder, L_{edge} and L_p loss help in the generation of a magnified frame of appropriate quality. As shown in Table 4, after the inclusion of all the modules and losses in the training process, the proposed method has the minimum MSE value.

5. Limitation

The dataset produced by Oh *et al.* [23] is used for training the proposed network. Since the dataset is synthetic (due to the unavailability of real ground truth), there is a domain gap. As hand-crafted method ignores fast large motion acceleration and jerk motion. Whereas in the dataset, the maximum input pixel displacement for magnification is up to 10 pixels. If objects with unwanted subtle motion like snow or rain *etc* come in this input range, they will also be magnified. Additionally, hand-crafted methods can also magnify color changes. But the SOTA deep learning methods are only for motion magnification (including ours). Hybrid approaches can be explored as an interesting area of research to close this domain gap. Also, deep learning methods produce some blur and texture smoothing for reducing the effects of noise. So, there is a gap between the base model and the lightweight model. More work needs to be done to further improve the lightweight model.

6. Conclusion

In this paper, we propose a deep learning based model for video motion magnification. It consists of proxy model based feature loss, feature sharing based encoders, and appearance encoder based regularization terms, to reduce the effects of noise, illumination *etc* and refine the motion features. The appearance encoder also helps to extract common appearance in the input frames, and combine it with the manipulator output, which is given to the decoder to produce a magnified frame. Additionally, a lightweight model with reduced computational complexity is proposed along with the base model. The results of the proposed models are evaluated qualitatively and quantitatively on real and synthetic videos with SOTA methods. Results show that the proposed models perform better than the SOTA methods both qualitatively and quantitatively for motion magnification.

References

[1] Miika Aittala and Frédo Durand. Burst image deblurring using permutation invariant convolutional neural networks. In

Proceedings of the European Conference on Computer Vision (ECCV), pages 731–747, 2018.

[2] Biagio Brattoli, Uta Büchler, Michael Dorkenwald, Philipp Reiser, Linard Filli, Fritjof Helmchen, Anna-Sophia Wahl, and Björn Ommer. Unsupervised behaviour analysis and magnification (ubam) using deep learning. *Nature Machine Intelligence*, 3(6):495–506, 2021.

[3] Justin G Chen, Neal Wadhwa, Young-Jin Cha, Frédo Durand, William T Freeman, and Oral Buyukozturk. Structural modal identification through high speed camera video: Motion magnification. In *Topics in Modal Analysis I, Volume 7*, pages 191–197. Springer, 2014.

[4] Weixuan Chen and Daniel McDuff. Deepmag: Source-specific change magnification using gradient ascent. *ACM Trans. Graph.*, 40(1), Sept. 2020.

[5] Abe Davis*, Katherine L. Bouman*, Justin G. Chen, Michael Rubinstein, Oral Büyüköztürk, Frédo Durand, and William T. Freeman. Visual vibrometry: Estimating material properties from small motions in video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):732–745, 2017.

[6] Michael Dorkenwald, Uta Buchler, and Bjorn Ommer. Unsupervised magnification of posture deviations across subjects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[7] Mohamed Elgharib, Mohamed Hefeeda, Fredo Durand, and Bill Freeman. Video magnification in presence of large motions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4119–4127, 2015.

[8] Wenkang Fan, Zhuohui Zheng, Wankang Zeng, Yinran Chen, Hui-Qing Zeng, Hong Shi, and Xiongbiao Luo. Robotically surgical vessel localization using robust hybrid video motion magnification. *IEEE Robotics and Automation Letters*, 6(2):1567–1573, 2021.

[9] Philipp Flotho, Cosmas Heiss, Gabriele Steidl, and Daniel J Strauss. Lagrangian motion magnification with double sparse optical flow decomposition. *arXiv preprint arXiv:2204.07636*, 2022.

[10] William T Freeman, Edward H Adelson, et al. The design and use of steerable filters. *IEEE Transactions on Pattern analysis and machine intelligence*, 13(9):891–906, 1991.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[12] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015.

[13] Guanzhe Hong, Zhiyuan Mao, Xiaojun Lin, and Stanley H Chan. Student-teacher learning from clean inputs to noisy inputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12075–12084, 2021.

[14] Xiaobin Hu, Wenqi Ren, Kaicheng Yu, Kaihao Zhang, Xiaochun Cao, Wei Liu, and Bjoern Menze. Pyramid architecture search for real-time image deblurring. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4298–4307, 2021.

[15] Rachel Huang, Jonathan Pedoeem, and Cuixian Chen. Yolo-lite: a real-time object detection algorithm optimized for non-gpu computers. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 2503–2510. IEEE, 2018.

[16] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

[17] Hyungjun Kim, Youngbeen Chung, Jie Jin, and Junhong Park. Manifestation of flexural vibration modes of rails by the phase-based magnification method. *IEEE Access*, 9:98121–98131, 2021.

[18] Julian FP Kooij and Jan C van Gemert. Depth-aware motion magnification. In *European Conference on Computer Vision*, pages 467–482. Springer, 2016.

[19] Ce Liu, Antonio Torralba, William T Freeman, Frédo Durand, and Edward H Adelson. Motion magnification. *ACM transactions on graphics (TOG)*, 24(3):519–526, 2005.

[20] Tambet Matiisen, Avital Oliver, Taco Cohen, and John Schulman. Teacher–student curriculum learning. *IEEE Transactions on Neural Networks and Learning Systems*, 31(9):3732–3740, 2020.

[21] Ernesto Moya-Albor, Jorge Brieva, Hiram Ponce, and Lourdes Martínez-Villaseñor. A non-contact heart rate estimation method using video magnification and neural networks. *IEEE Instrumentation Measurement Magazine*, 23(4):56–62, 2020.

[22] Ewa M. Nowara, Daniel McDuff, and Ashok Veeraraghavan. Combining magnification and measurement for non-contact cardiac monitoring. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3805–3814, 2021.

[23] Tae-Hyun Oh, Ronnachai Jaroensri, Changil Kim, Mohamed Elgharib, Fr'edo Durand, William T Freeman, and Wojciech Matusik. Learning-based video motion magnification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 633–648, 2018.

[24] Sylvain Paris, Samuel W Hasinoff, and Jan Kautz. Local laplacian filters: edge-aware image processing with a laplacian pyramid. *Communications of the ACM*, 58(3):81–91, 2015.

[25] Cong Peng, Cong Zeng, and Yangang Wang. Phase-based noncontact vibration measurement of high-speed magnetically suspended rotor. *IEEE Transactions on Instrumentation and Measurement*, 69(7):4807–4817, 2020.

[26] Vincent Perrot, Sébastien Salles, Didier Vray, and Hervé Liebgott. Video magnification applied in ultrasound. *IEEE Transactions on Biomedical Engineering*, 66(1):283–288, 2019.

[27] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[28] Shoichiro Takeda, Yasunori Akagi, Kazuki Okami, Megumi Isogai, and Hideaki Kimata. Video magnification in the wild using fractional anisotropy in temporal distribution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1614–1622, 2019.

[29] Shoichiro Takeda, Kenta Niwa, Mariko Isogawa, Shinya Shimizu, Kazuki Okami, and Yushi Aono. Bilateral video magnification filter. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17369–17378, June 2022.

[30] Shoichiro Takeda, Kazuki Okami, Dan Mikami, Megumi Isogai, and Hideaki Kimata. Jerk-aware video acceleration magnification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1769–1777, 2018.

[31] Manisha Verma and Shanmuganathan Raman. Interest region based motion magnification. In *International Conference on Image Analysis and Processing*, pages 27–39. Springer, 2017.

[32] Manisha Verma and Shanmuganathan Raman. Edge-aware spatial filtering-based motion magnification. In *Proceedings of 2nd International Conference on Computer Vision & Image Processing*, pages 117–128. Springer, 2018.

[33] Neal Wadhwa, Michael Rubinstein, Frédo Durand, and William T Freeman. Phase-based video motion processing. *ACM Transactions on Graphics (TOG)*, 32(4):1–10, 2013.

[34] Neal Wadhwa, Michael Rubinstein, Frédo Durand, and William T Freeman. Riesz pyramids for fast phase-based video magnification. In *2014 IEEE International Conference on Computational Photography (ICCP)*, pages 1–10. IEEE, 2014.

[35] Zhou Wang and Eero P Simoncelli. An adaptive linear system framework for image distortion analysis. In *IEEE International Conference on Image Processing 2005*, volume 3, pages III–1160. IEEE, 2005.

[36] Hao-Yu Wu, Michael Rubinstein, Eugene Shih, John Guttag, Frédo Durand, and William Freeman. Eulerian video magnification for revealing subtle changes in the world. *ACM transactions on graphics (TOG)*, 31(4):1–8, 2012.

[37] Xiu Wu, Xuezhi Yang, Jing Jin, and Zhao Yang. Pca-based magnification method for revealing small signals in video. *Signal, Image and Video Processing*, 12(7):1293–1299, 2018.

[38] Zongsheng Yue, Jianwen Xie, Qian Zhao, and Deyu Meng. Semi-supervised video deraining with dynamical rain generator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 642–652, 2021.

[39] Yichao Zhang, Silvia L Pintea, and Jan C Van Gemert. Video acceleration magnification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 529–537, 2017.