# Learning-based Video Motion Magnification

Asma Boukhdhir, Edgard Dabier, Maria Florenza, Mohammad Dolati, Rayane Hamdadou

# Summary

1. Motion Magnification
2. Prior work
3. Learning based Motion Magnification
4. Methodology and Dataset
5. Results Analysis
6. Questions

# 1. Motion magnification

- What is Motion Magnification?
- Applications
- Two Main Approaches
- Eulerian Perspective
- Temporal Filtering

# What is Motion Magnification?


Input video (left) and 20x magnified motion video (right)

**Definition :** Motion magnification allows one to **see small motions** in videos that are normally invisible to the human eye, by **magnifying** them.

# Applications


**Structural Health Monitoring**


**Medical and Biomedical Applications**

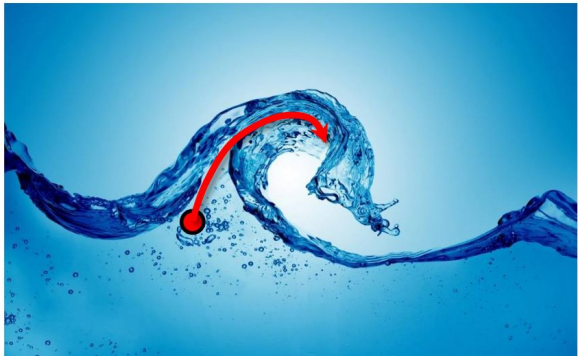

**Industrial Diagnostics**

# How to extract motion information ?

## Lagrangian approach

Motion is defined as **displacement** of objects from one frame to another.

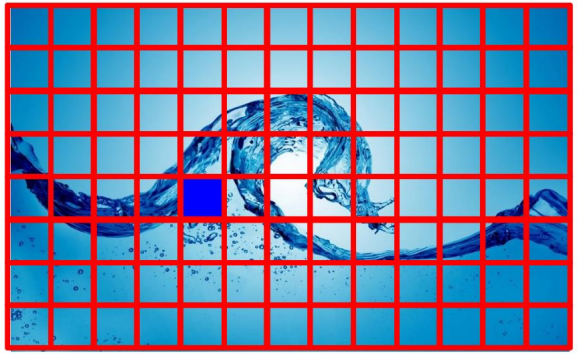
Track individual objects through time



## Eulerian approach

Motion is defined as changes in **intensity** at fixed pixel positions over time.

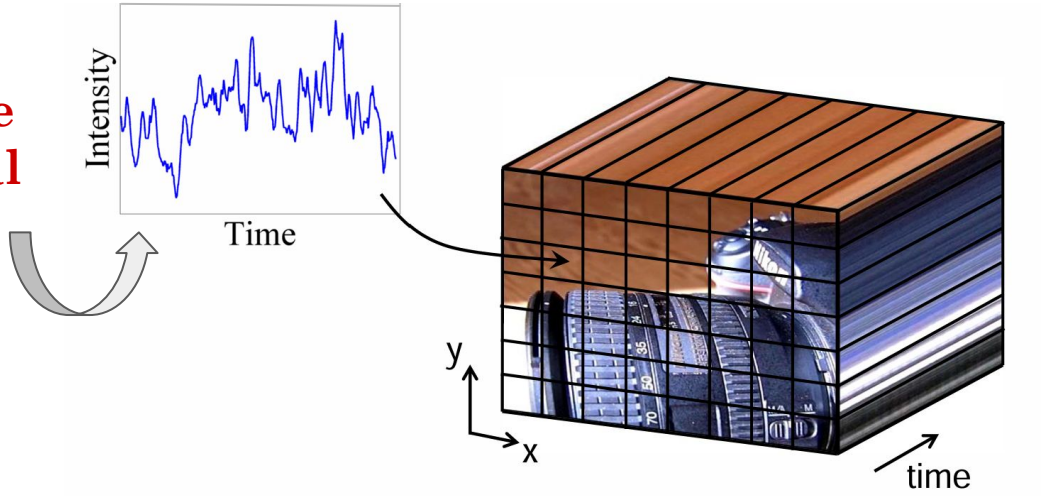
Track intensity changes over time at each pixel



# Eulerian Perspective

- Each pixel is processed independently


- Treat each pixel as **a time series and apply signal processing to it**



# Eulerian Motion Magnification Problem Setup

Given an image

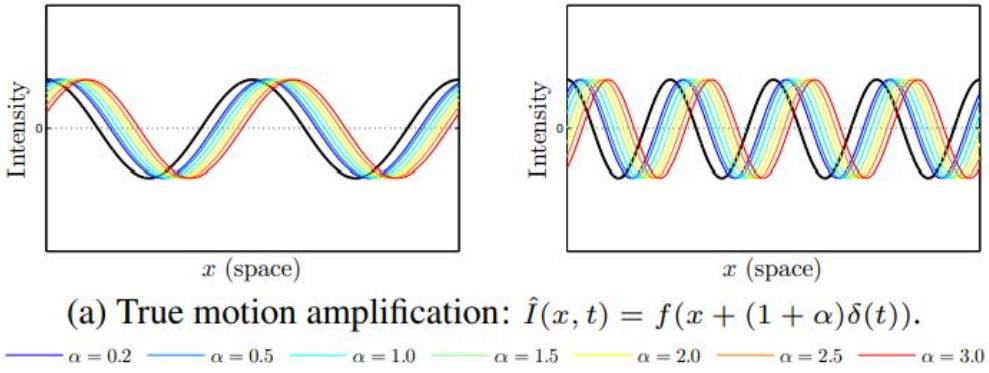

$$I(\mathbf{x}, t) = f(\mathbf{x} + \delta(\mathbf{x}, t))$$

Motion field

Goal of magnification

Motion Magnified image $\tilde{I}(\mathbf{x}, t) = f(\mathbf{x} + (1 + \alpha)\delta(\mathbf{x}, t))$


Magnification factor

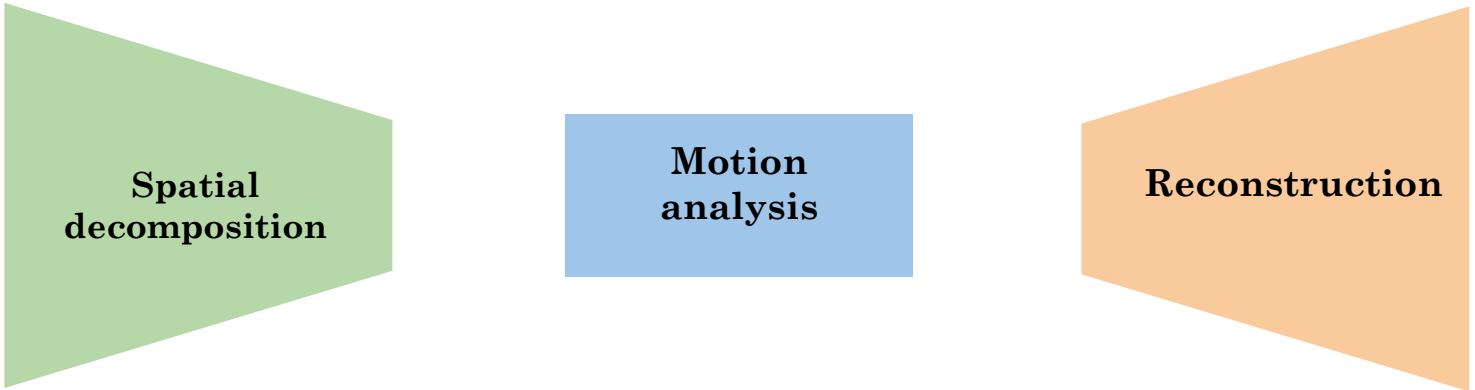


# 2. Prior work

- Eulerian Linear Motion Magnification
- Eulerian Phase-based Motion Magnification
- Limitations

# Generic Architecture of Eulerian Motion Magnification Framework

$$I(\mathbf{x},t) = f(\mathbf{x}+\delta(\mathbf{x},t)) \quad \Longrightarrow \quad \delta(\mathbf{x},t) \quad \Longrightarrow \quad \tilde{I}(\mathbf{x},t) = f(\mathbf{x}+(1+\alpha)\delta(\mathbf{x},t))$$




Decompose video frames into **representations** that allow motion analysis.

Extract the motion field from these representations

Reconstruct the video using motion-magnified frames

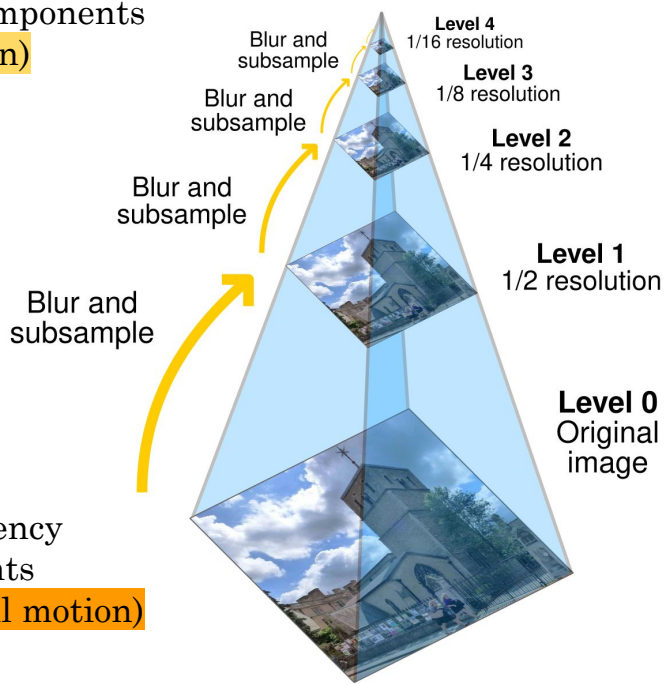
| Method | Liu *et al.* [13] | Wu *et al.* [27] | Wadhwa *et al.* [24] | Wadhwa *et al.* [25] | Zhang *et al.* [28] | Ours |
|---|---|---|---|---|---|---|
| **Spatial decomposition** | Tracking, optical flow | Laplacian pyramid | Steerable filters | Riesz pyramid | Steerable filters | Deep convolution layers |
| **Motion isolation** | - | Temporal bandpass filter | Temporal bandpass filter | Temporal bandpass filter | Temporal bandpass filter (2nd-order derivative) | Subtraction or temporal bandpass filter |
| **Representation denoising** | Expectation-Maximization | - | Amplitude weighted Gaussian filtering | Amplitude weighted Gaussian filtering | Amplitude weighted Gaussian filtering | Trainable convolution |

**Table 1. Comparisons of the prior arts.**

# Representations result from hand-designed filters !

Image pyramids

Low frequency components
(large scale motion)



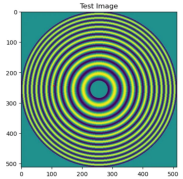

High frequency components
(high-detail, small motion)

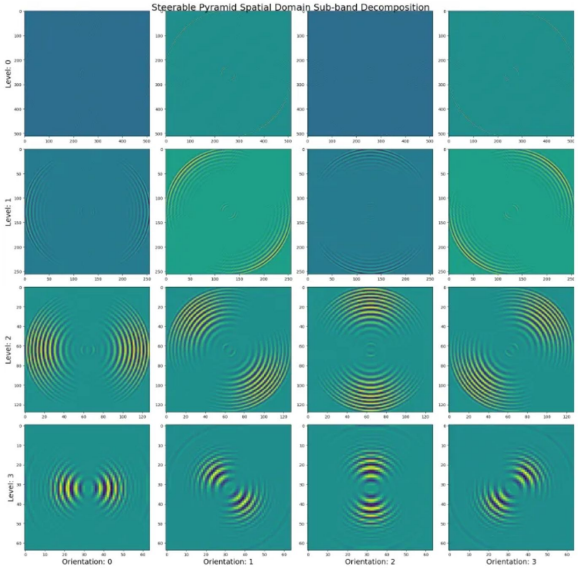
Multi-scale decomposition

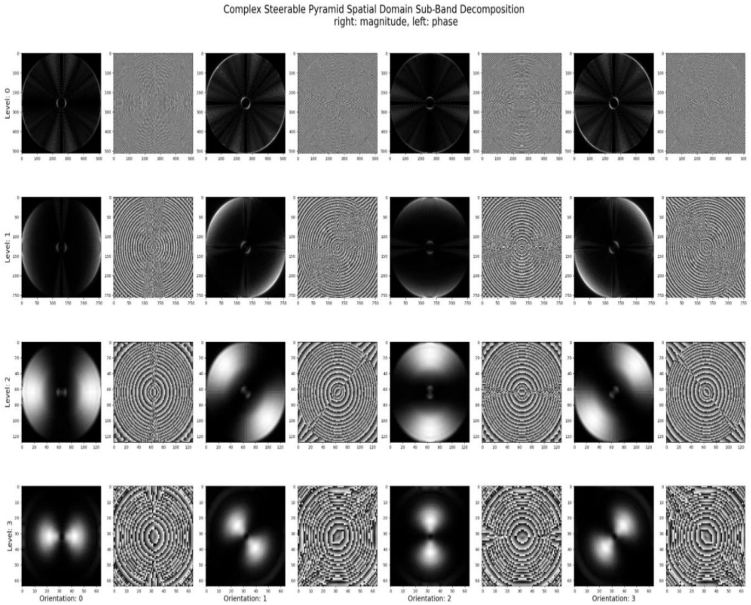
# Representations result from hand-designed filters !

A steerable pyramid

A complex steerable pyramid



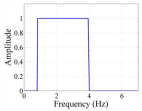

Multi-scale and multi-orientation decomposition

Multi-scale and multi-orientation decomposition + Local phase information

# **Prior Work** - Eulerian Linear motion magnification

1st order Taylor approximation

$$I(x,t) \approx f(x) + \delta(t)\frac{\partial f(x)}{\partial x}$$




Let B(x, t) be the result of applying **a broadband temporal bandpass filter** to I(x, t) at every position x

$$B(x,t) = \delta(t)\frac{\partial f(x)}{\partial x}.$$

✖ α
Mag factor

$$\tilde{I}(x,t) \approx f(x) + (1+\alpha)\delta(t)\frac{\partial f(x)}{\partial x}.$$

Assuming the first-order Taylor expansion holds for the amplified larger perturbation, (1 + α)δ(t)

$$\tilde{I}(x,t) \approx f(x + (1+\alpha)\delta(t)).$$

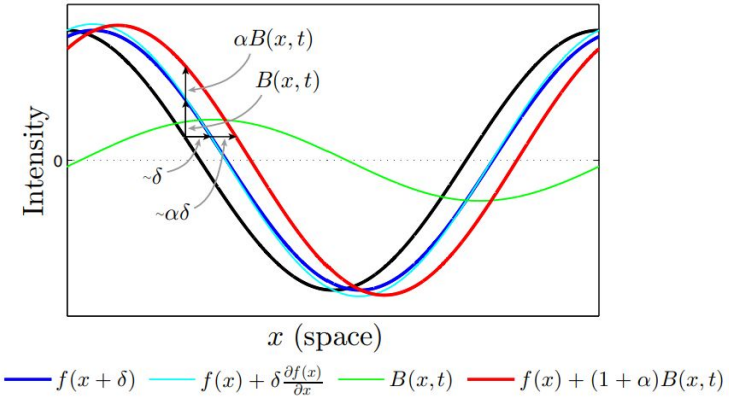
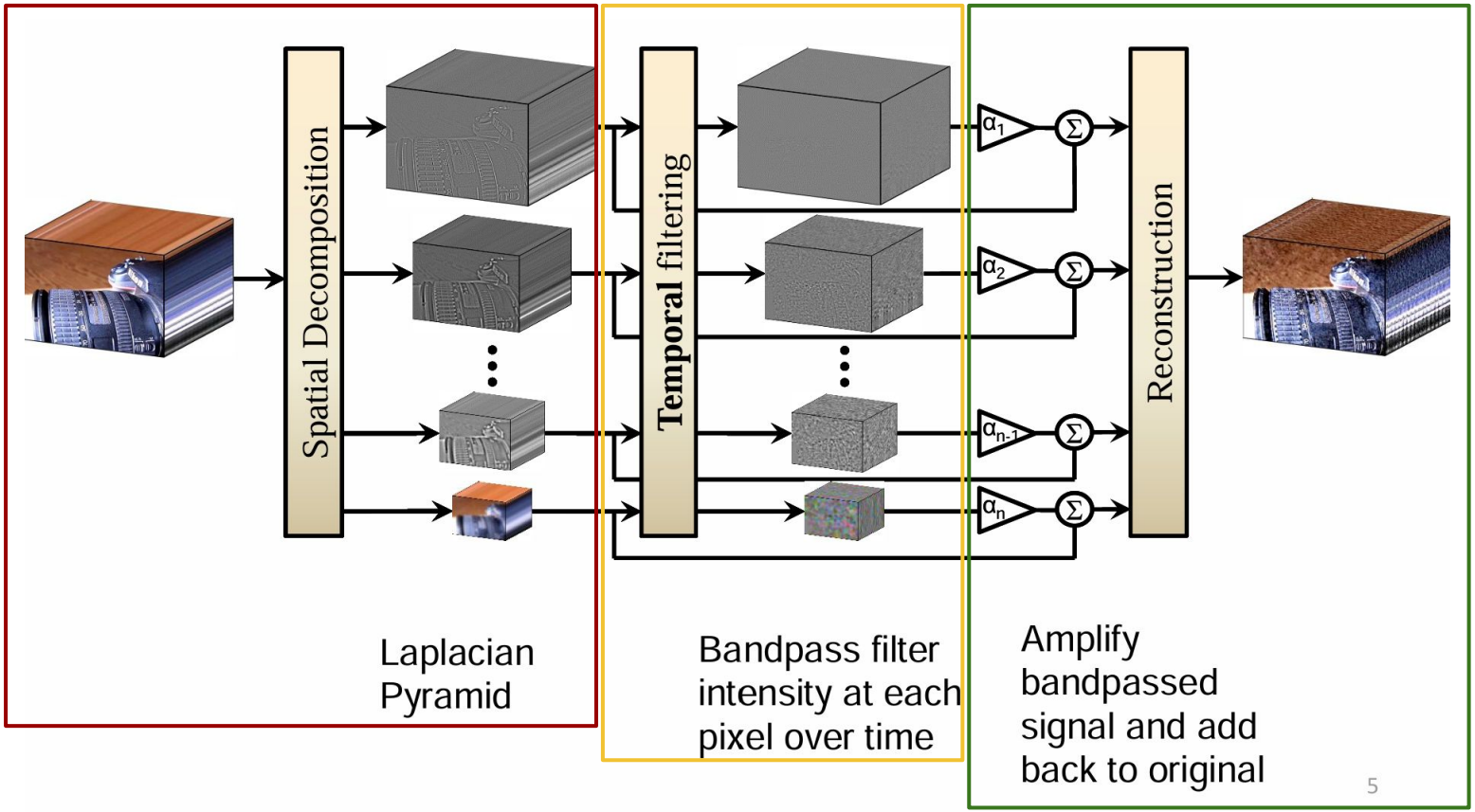



**Figure 3:** *Temporal filtering can approximate spatial translation. This effect is demonstrated here on a 1D signal, but equally applies to 2D. The input signal is shown at two time instants:* $I(x,t) = f(x)$ *at time* $t$ *and* $I(x,t+1) = f(x+\delta)$ *at time* $t+1$*. The first-order Taylor series expansion of* $I(x,t+1)$ *about* $x$ *approximates well the translated signal. The temporal bandpass is amplified and added to the original signal to generate a larger translation. In this example* $\alpha = 1$*, magnifying the motion by* $100\%$*, and the temporal filter is a finite difference filter, subtracting the two curves.*

**Motion magnification = changing intensities**

## Prior Work

# Eulerian Video Magnification framework

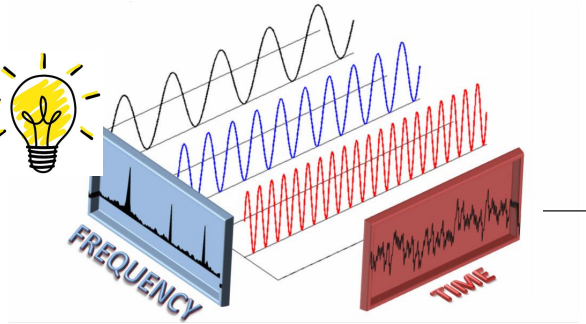


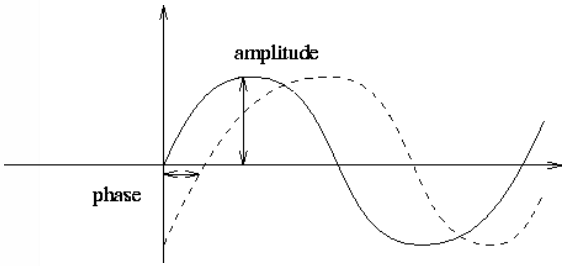
# **State-of-the-art** - Eulerian Phase-based motion magnification

## Approach in 1D

$$f(x+\delta(t)) = \sum_{\omega=-\infty}^{\infty} A_\omega e^{i\omega(x+\delta(t))}$$

$$\omega(x+\delta(t)) = \omega x + \omega\delta(t)$$




The phase of each sinusoid contains motion information

- $\omega x$ DC component = underlying spatial signal = The static background ⇨ Estimated from a reference frame
- $\omega\delta(t)$ component containing the motion information to extract

⚠ To isolate certain motion of interest (specific frequency band), a **temporal bandpass filter** is used

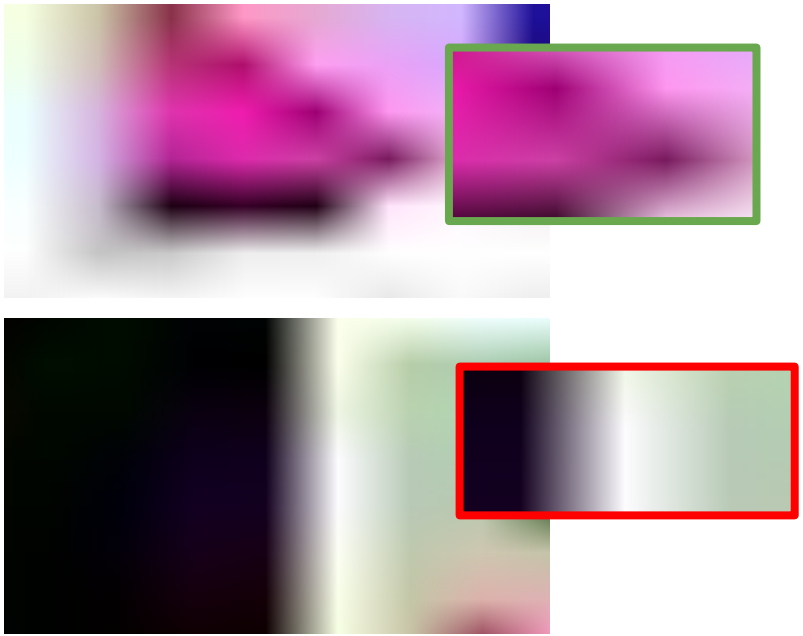
⇨ **Motion magnification = shifting phase**

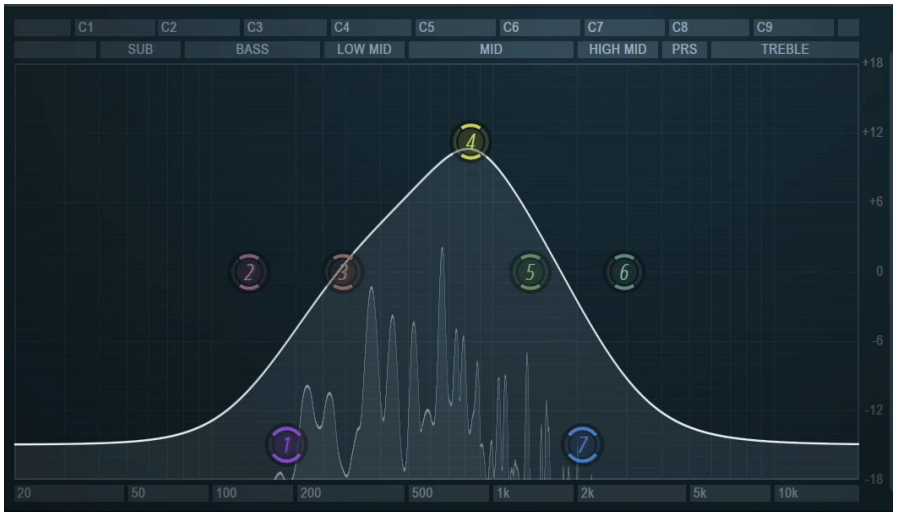
# Temporal Filtering

Temporal filtering/ bandpassing ⇔ Spectral filtering/ bandpassing

→ Movement selection

→ Sound selection

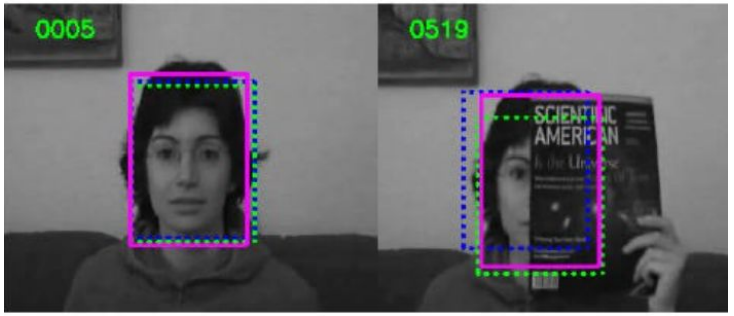




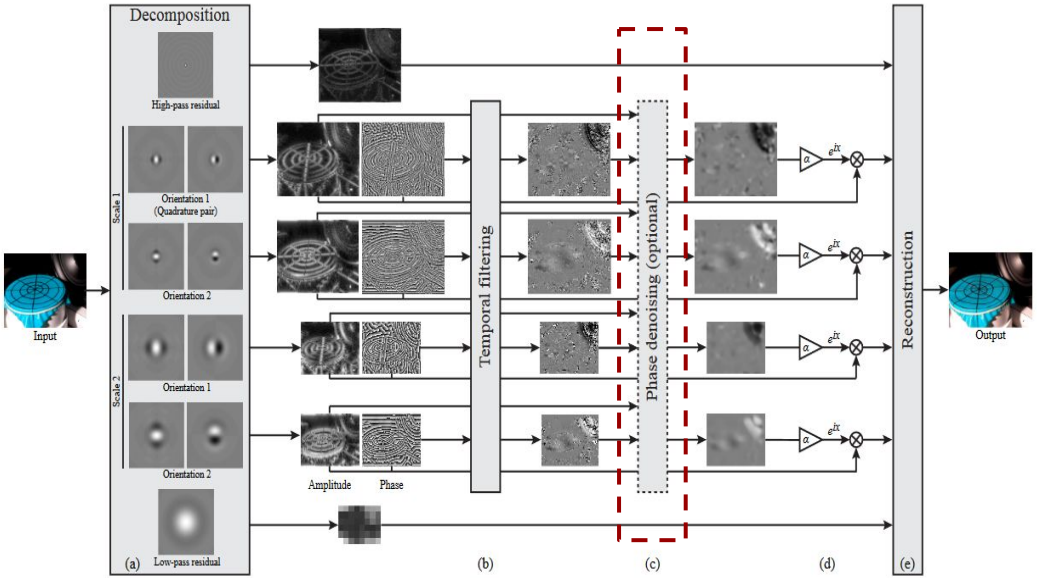
## Limitations

- Rely on hand-designed filters

- Do not take into account many issues such as occlusion



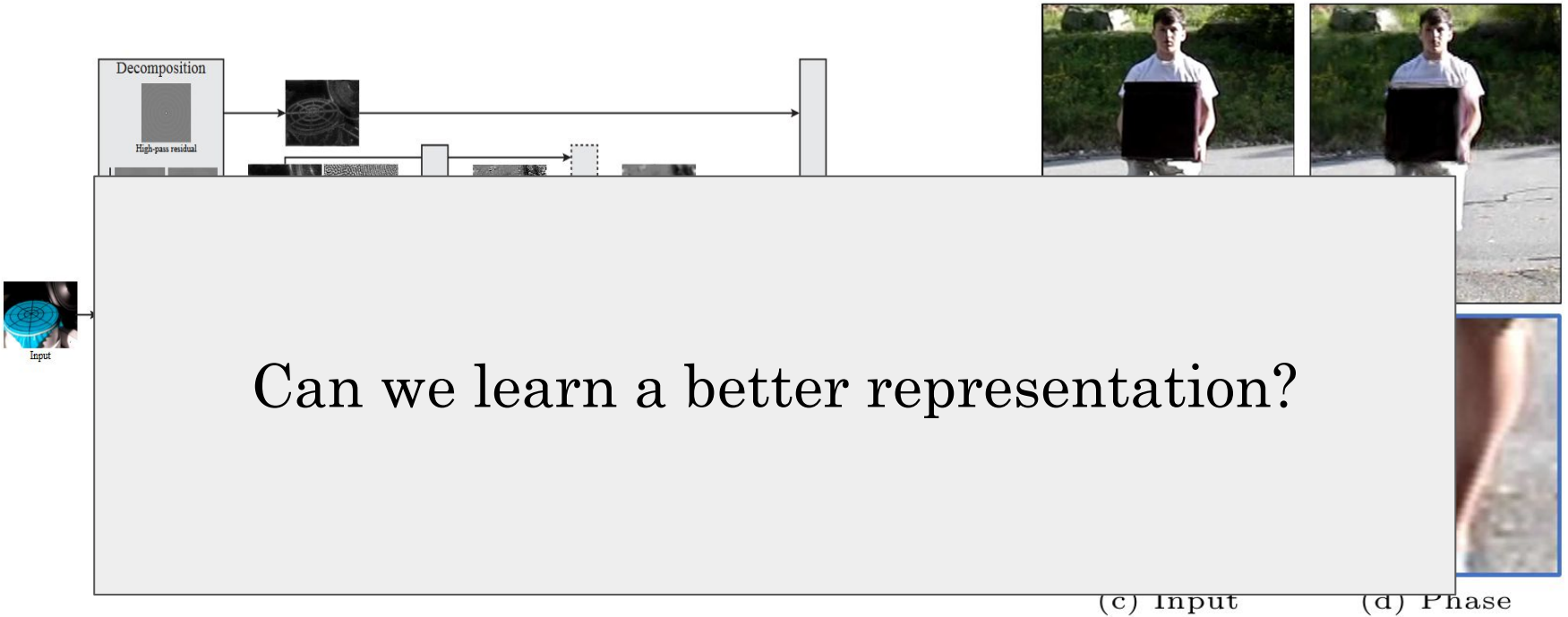
# Limitations




Prone to noise


Excessive blurring

# Limitations




Can we learn a better representation?

Prone to noise

Excessive blurring

# 3. Learning based motion magnification

- Why use Deep Learning?
- Challenges of Learning
- Model Architecture
- Learning Shape and Texture Representation

# Why use Deep Learning?

## Problems of traditional methods

- Noisy results
- Excessive blurring
- Manually tuned parameters

## Why deep learning is better?

- Learns motion representations automatically
- Better at separating motion from noise
- More adaptable to different applications

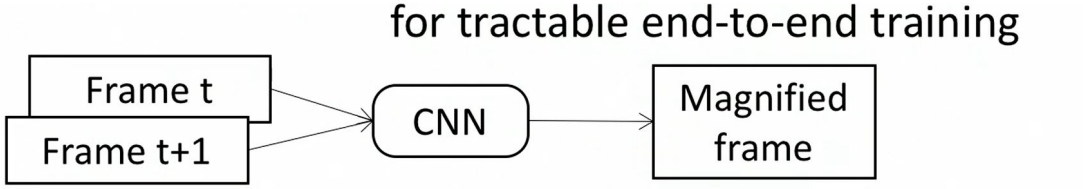
**Original** **Traditional method** **Deep Learning**

## Challenges of learning-based Motion Magnification

- Lack of real-world datasets
- Motion vs. noise separation
- Capturing motion history is computationally expensive

# Challenges of learning-based Motion Magnification

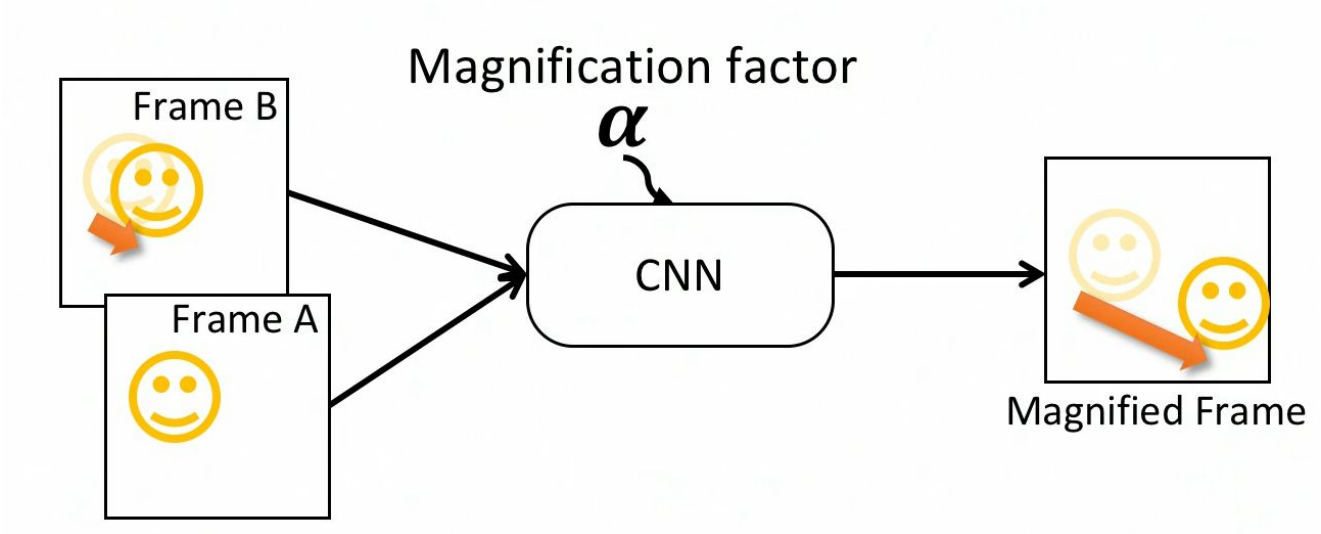
- **Temporal complexity** → Uses **2-frame-based training** instead of multi-frame sequences → computationally feasible




- **Motion vs. noise separation** → Replaces traditional hand-crafted filters with deep learning→ **CNN learns** to extract and manipulate motion representations with a magnification factor $\alpha$.

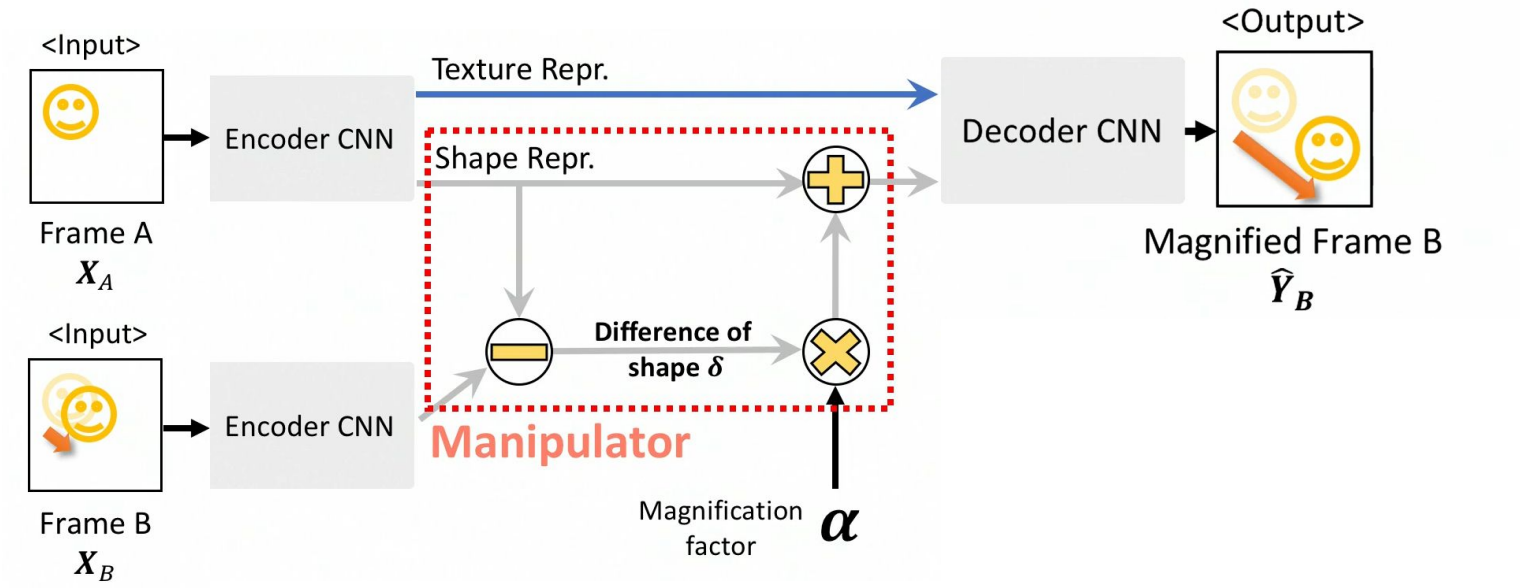
- **Lack of real-world data** → Solved with **synthetic data generation.**

# Two-Frames Magnification Framework




- Easy and efficient to train end-to-end network
- Easy to generate synthetic data and model CNN architecture

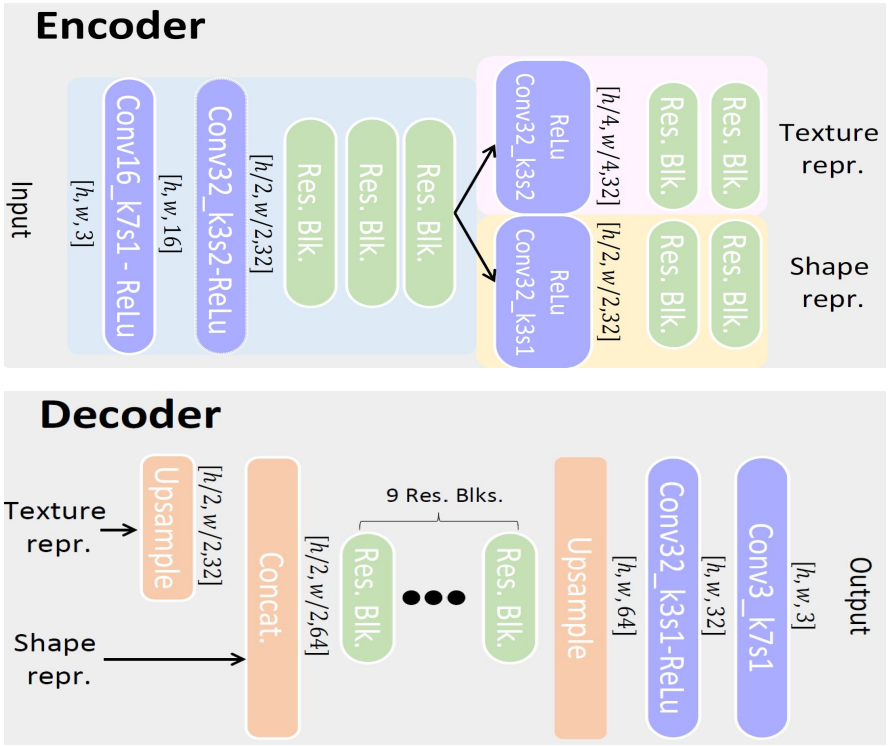
# Model Architecture



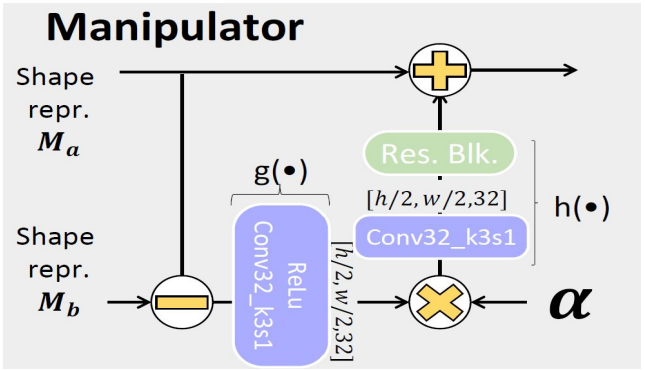

Three main components:

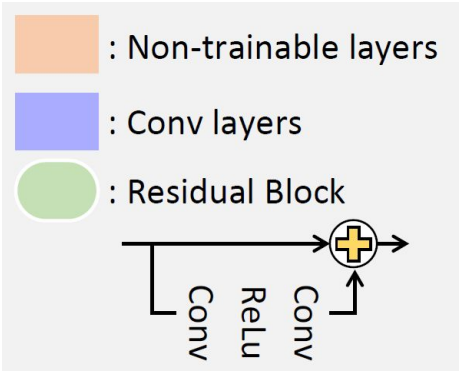
- **Encoder Ge(·):** **Extracts shape & texture** representations.
- **Manipulator Gm(·):** **Computes and magnifies** the motion.
- **Decoder Gd(·):** Combines shape & texture to **reconstruct** the output magnified frame.

# **Model Architecture** - Detailed Network Architecture




$$G_m(\mathbf{M}_a, \mathbf{M}_b, \alpha) = \mathbf{M}_a + h\left(\alpha \cdot g(\mathbf{M}_b - \mathbf{M}_a)\right)$$



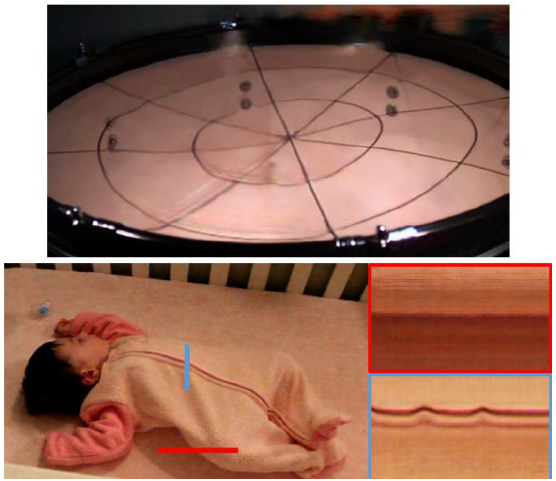
## **Model Architecture** - Linear vs. Non-Linear Manipulation

### Linear

Simple multiplication of motion difference, but **blurs edges and amplifies noise**.

$$G_m(\mathbf{M}_a, \mathbf{M}_b, \alpha) = \mathbf{M}_a + \alpha(\mathbf{M}_b - \mathbf{M}_a)$$

Linear

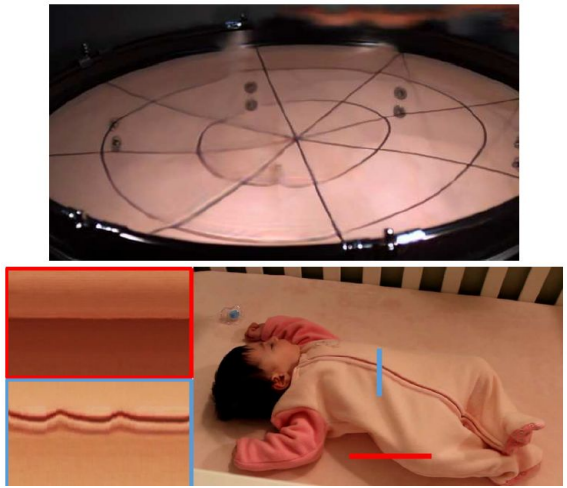


### Non-Linear

Adds convolutional layers and ReLU, **improving edge preservation and noise robustness**.

$$G_m(\mathbf{M}_a, \mathbf{M}_b, \alpha) = \mathbf{M}_a + h\left(\alpha \cdot g(\mathbf{M}_b - \mathbf{M}_a)\right)$$

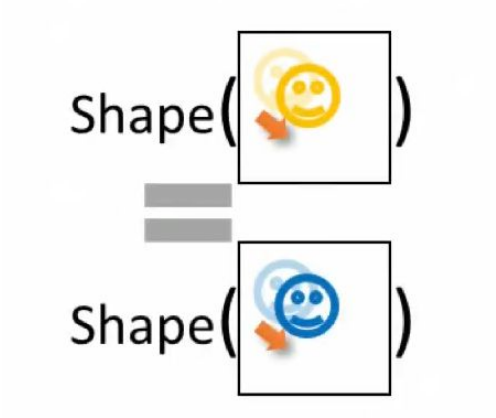
Non-Linear



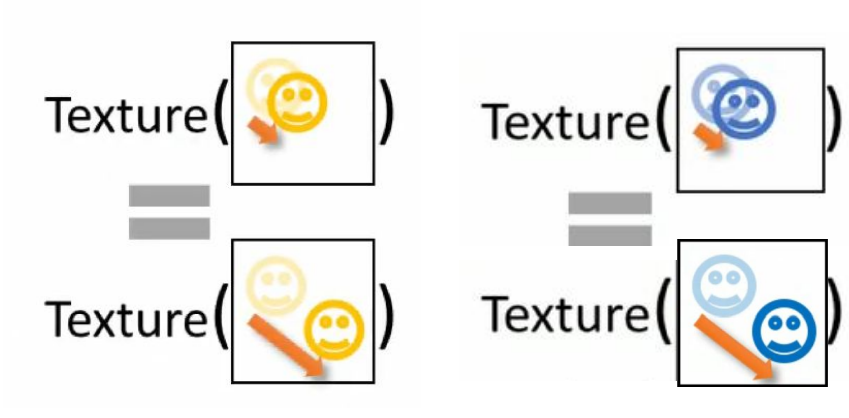
## **Learning Shape and Texture Representations** - In the architecture

Idea : Use **Color perturbation**


### Shape regularization




### Text regularization



## **Learning Shape and Texture Representations** - In the loss function

$$\mathcal{L}_1(\mathbf{Y}, \hat{\mathbf{Y}}) + \lambda(\mathcal{L}_1(\mathbf{V}_a, \mathbf{V}_b) + \mathcal{L}_1(\mathbf{V}'_b, \mathbf{V}'_Y) + \mathcal{L}_1(\mathbf{M}_b, \mathbf{M}'_b))$$

**Loss term**

**Regularization terms** to drive the separation of the texture and the shape representations.

Y: Ground-truth magnified-frame

Ŷ: Predicted output

M: Shape representation

V: Texture representation

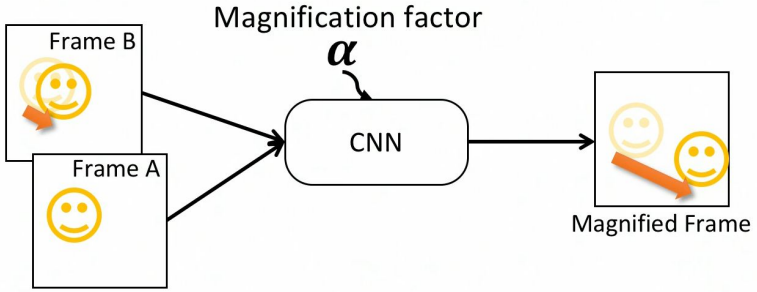
The prime symbol indicates color perturbation

# 4. Methodology and Dataset

- 2-Frame Setting
- Temporal Operation
- Synthetic Dataset

# **Methodology and Dataset** - 2-Frame Setting

Temporal aspect of motion is ignored →



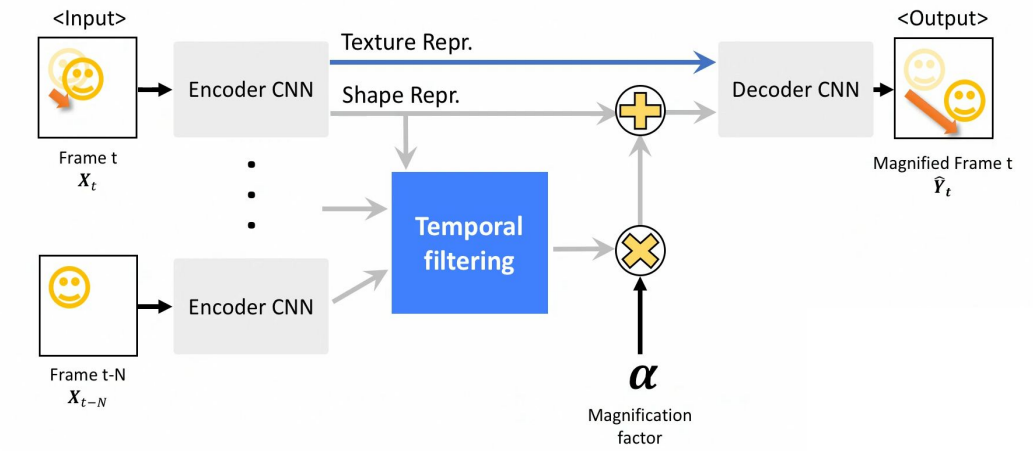

| Mode | Reference | What is magnified | Interpretation of $\alpha$ |
|---|---|---|---|
| **Static mode** | First frame (fixed) | Motion relative to a fixed reference | Magnifies displacement |
| **Dynamic mode** | Previous frame $(x_{t-1})$ | Difference between frames (velocity) | Magnifies velocity |

## **Methodology and Dataset** - Temporal Operation

High magnification = Noise & undesired motion !

The shape representation M(t) is linear enough w.r.t . the displacement → compatible with linear temporal filters

$$G_{m,\,temporal}(M(t), \alpha) = M(t) + \alpha\, T(M(t))$$




M(t): shape representation

T(·): pixel-wise temporal filter

$\alpha$: magnification factor

## **Methodology and Dataset** - Synthetic Dataset

Why Synthetic Data?

- Hard to obtain real motion-magnified frame pairs
- Synthetic data enables controlled, large-scale training

Dataset Composition:

- Background: images from MS COCO
- Foreground: segmented objects from PASCAL VOC
- Training Samples: 7-15 objects

## **Methodology and Dataset** - Synthetic Dataset

Challenge: Low contrast and small motion → Add additional examples

- Low Contrast: Blurred backgrounds and moving background only
- Small Motion: Static scenes and moving foreground with static background

→ Better performance on low contrast and small motion

Dataset:

- 5 parts, 20,000 samples each
- 384x384 images

# **Methodology and Dataset** - Synthetic Dataset

## Input motion and Amplification Factor($\alpha$)

Max 10px input motion

Max 100x magnification

Max 30px magnified motion

## Subpixel Motion Generation


Translation

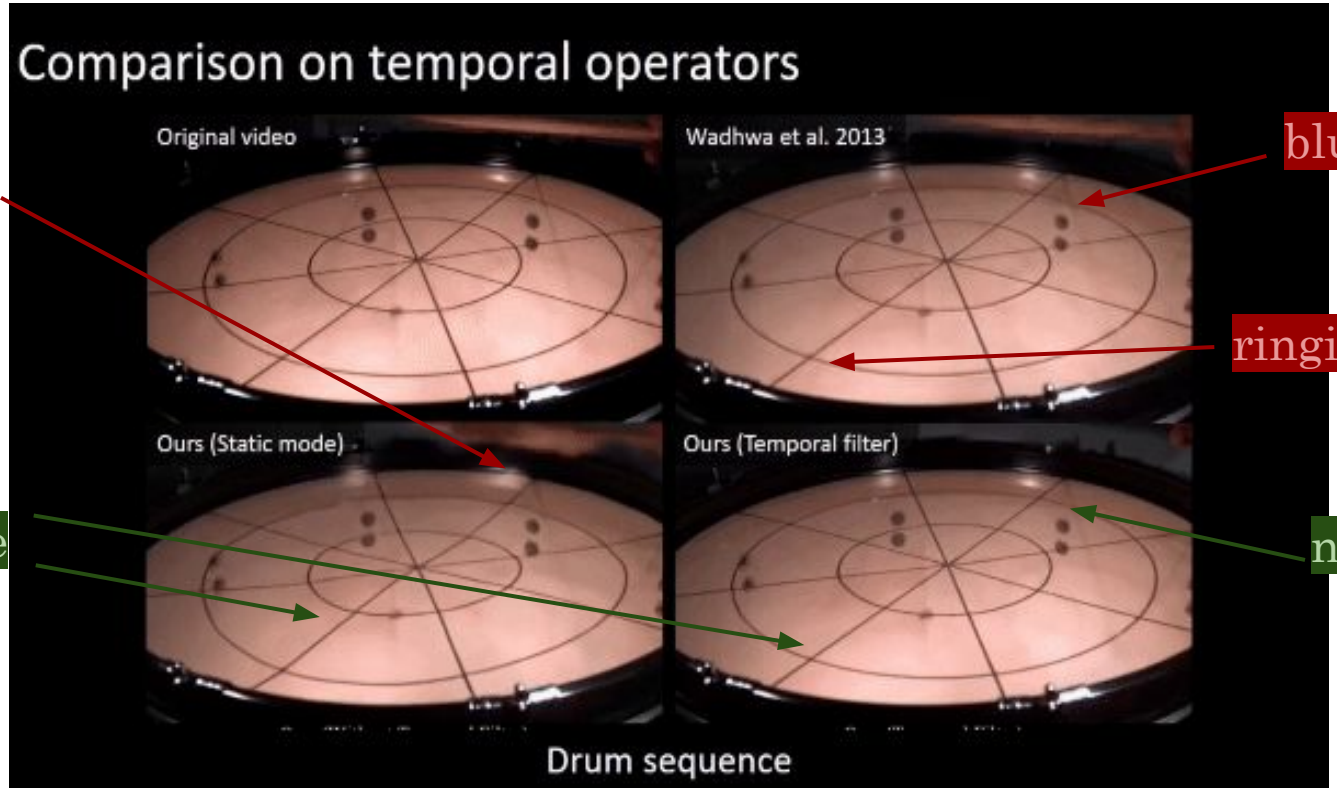
Gaussian filter + Bicubic interpolation

Uniform quantization noise added

# 5. Results Analysis

- Comparison with the state of the art
- Physical accuracy
- Visualization and explainability
- Limitations

## **Comparison with the state of the art** - static and temporal filtered




blurring

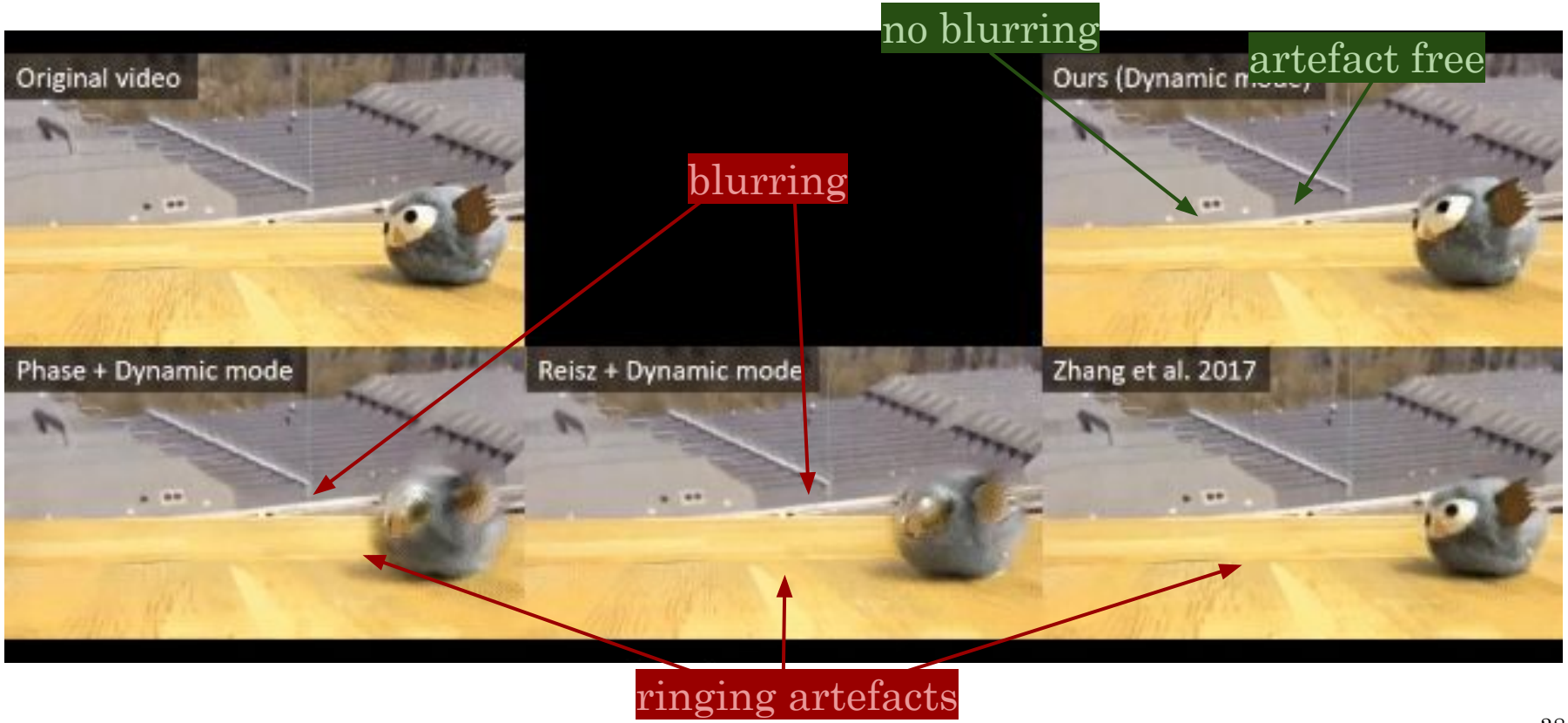
blurring

ringing artefact

artefact free

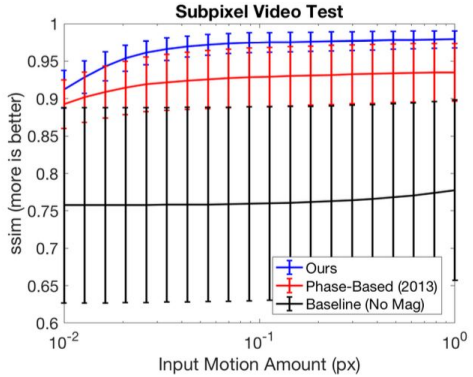
no blurring

## **Comparison with the state of the art -** dynamic mode



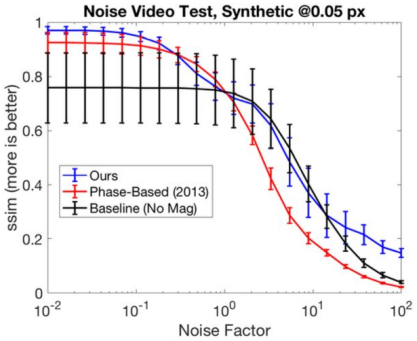
# **Comparison with the state of the art -** Small motion and noise robustness

**Motion test**: no noise, mag. factor $\alpha \nearrow$ to have 10px motion

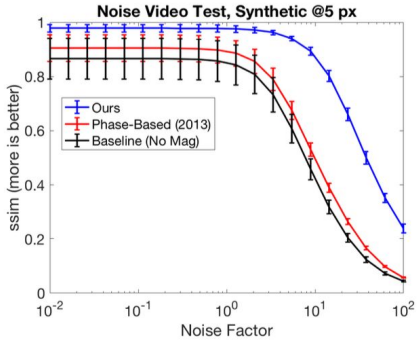



(a) Sub-pixel motion performance

**Noise test**: input motion and mag. factor fixed




(b) Noise performance with small input motion (0.05 px)




(c) Noise performance with large input motion (5 px)

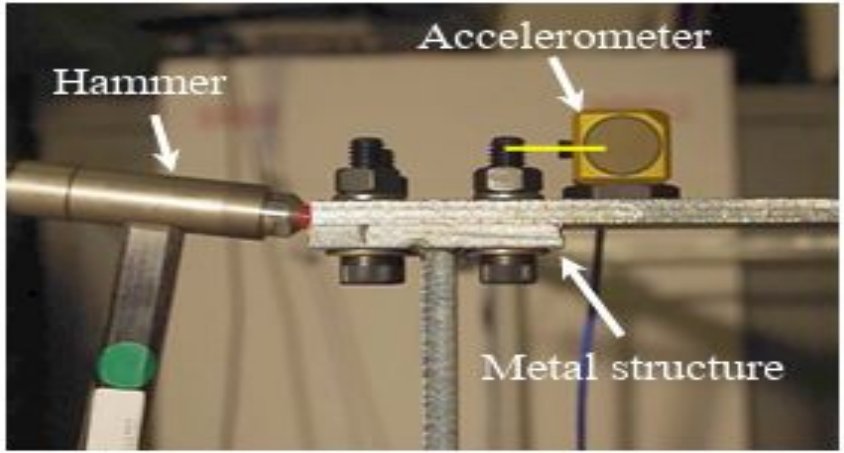
Smaller motion

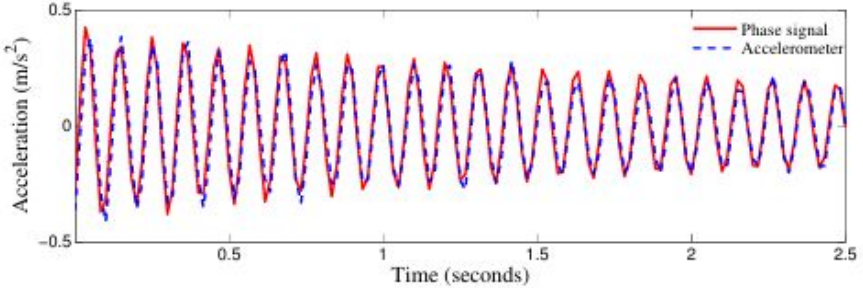
Small motions = harder to distinguish from noise

Better results than phase based methods.

*Baseline (SSIM between input and output frames) for reference

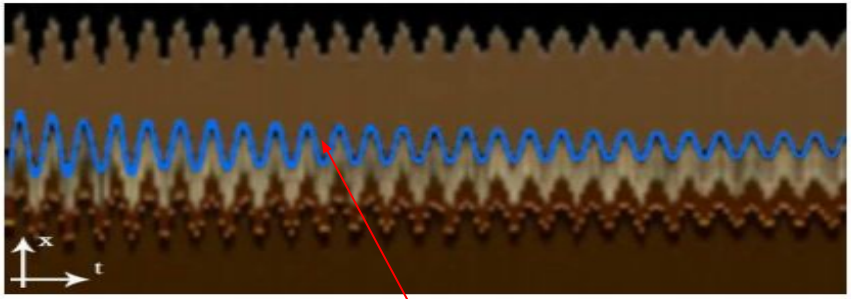
# **Model's physical Accuracy** - Hammer sequence analysis



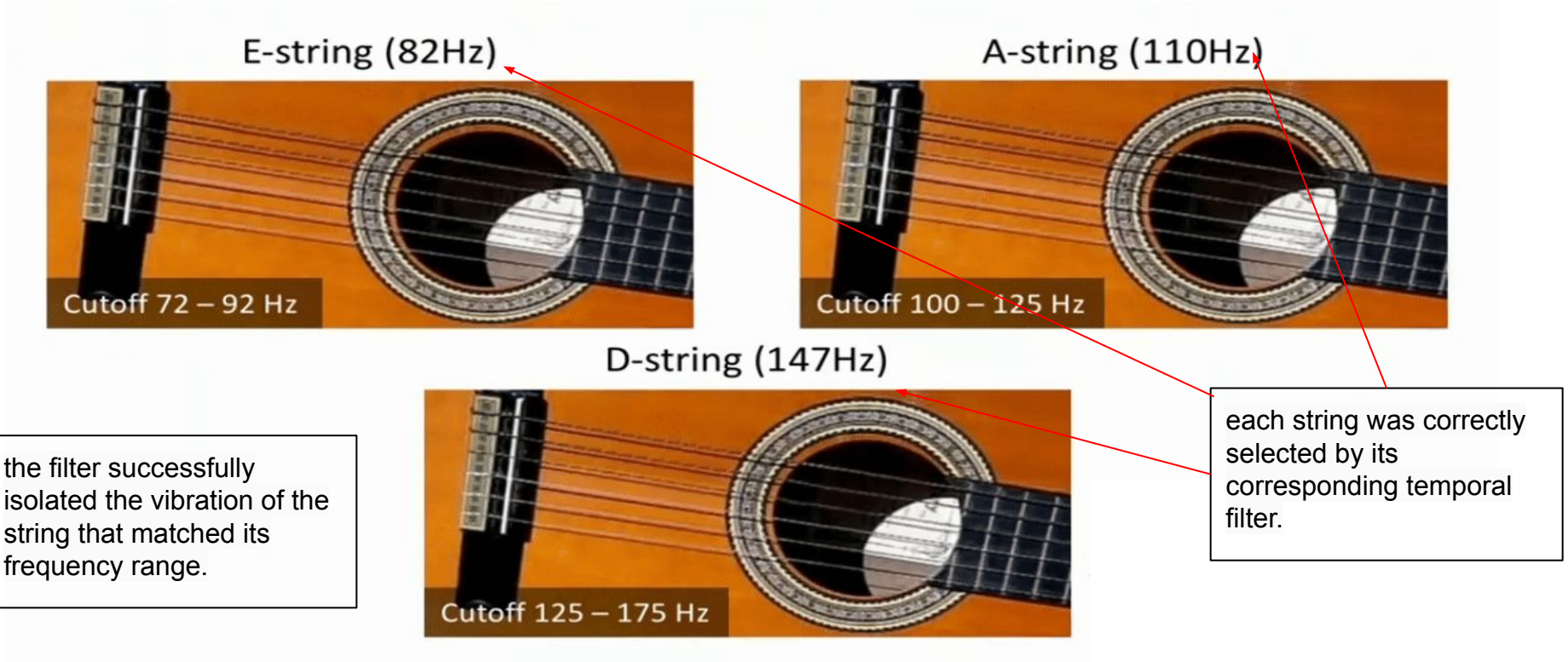

A comparison of acceleration extracted from the video with the accelerometer recording.

1. **Integrating the Signal Twice**: converting the acceleration data into displacement data (the actual position change over time)
2. **Zero-Phase High-Pass Filter**: helps isolate the true motion caused by the hammer strike insuring meaningful vibration signals are left.




Magnified frame matches the accelerometer signal

# **Model's physical Accuracy** - Guitar sequence analysis



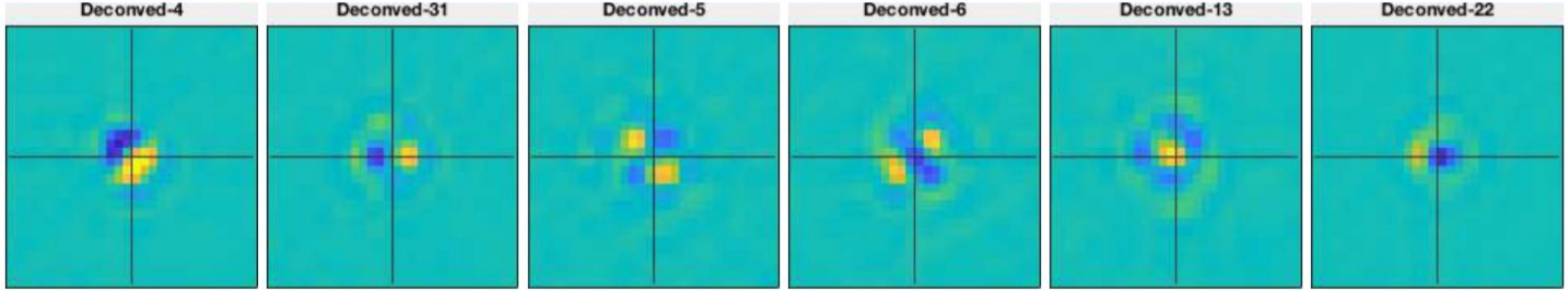

the filter successfully isolated the vibration of the string that matched its frequency range.

each string was correctly selected by its corresponding temporal filter.

Shape Representation is **linear enough**, the motion remains well-structured, allowing the filter to enhance subtle movements without introducing artifacts.

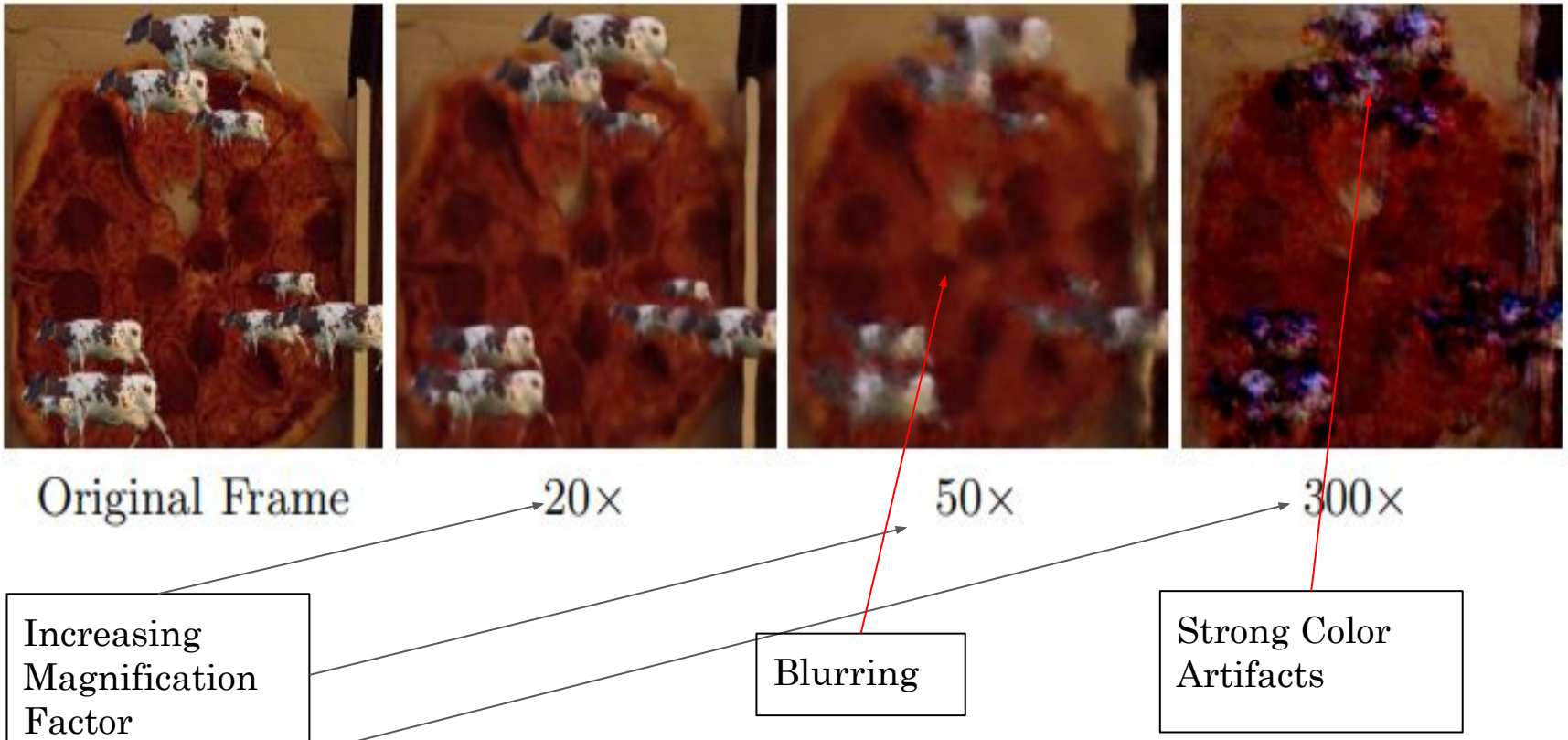
# Visualization and explainability

**Shape Branch**




| Gabor-like filters | Laplacian-like filters | Corner detector-like filters |
|---|---|---|
| Detect specific frequency content in an image | Identifying areas of rapid intensity change in an image | Identify corners or points where two edges meet at an angle |
| Capturing directional features | Capturing fine details and enhancing the clarity of edges | Provide stable points for tracking motion |

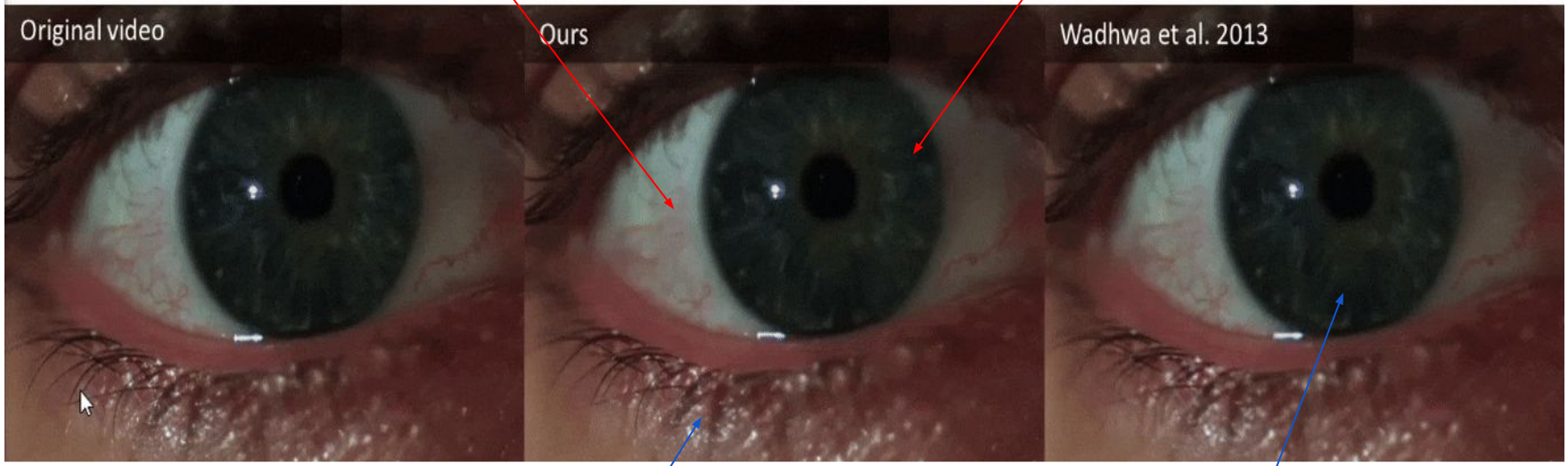
**Limitations** - high magnification factors



**Limitations** - very small motions ( eye sequence)

Artifacts

Patchy Magnification




Problem with the temporal filter!

little motion or no motion except in certain occasions

Able to reveal this small motion on the iris

**Limitations** - very small motions

Training on multiple frames could improve performance by allowing the model to better understand and capture the temporal aspects of motion


Some of the motions come back even though the edges appear to be noisy here !


Thank you
for listening

# 6. Questions

# Questions

1. What is the purpose of the temporal filters, and how do they interact with the learned representations?

2. How is the model learning to differentiate the shape representation and the texture representation of the image?

3. How does the system differentiate between large and small movements ?

## 1 What is the purpose of the temporal filters, and how do they interact with the learned representations?

- Temporal filters allow **the extraction of motion signals** from the learned representations.
- Additionally, specific parameters of the filters allow **the selection of certain frequency bands**, preventing the magnification of unwanted motion.

# 2 How is the model learning to differentiate the shape representation and the texture representation of the image?

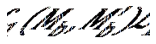
**1. Network Architecture (Explicit Separation in the Encoder):**
**Encoder splits into two branches**:

- **Shape Representation (M)** → Captures **motion-related features (edges, object boundaries, contours)**.
- **Texture Representation (V)** → Captures **appearance-based features (color, lighting, fine textures)**.

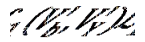
**Texture is downsampled more than shape** to remove **high-frequency noise** (like shadows or color variations) that could interfere with motion detection.

**2. Loss Function (Enforcing Separation Using Color Perturbation):**

- **Shape Regularization**:

 forces shape representation to remain unchanged, even when colors change

- **Text Regularization:**

 forces texture representation to adapt to color perturbation, ensuring texture captures only appearance changes, not motion

- **By penalizing incorrect feature mixing, the network learns to store shape and texture separately.**

## 3 How does the system differentiate between large and small movements?

- **Training on Synthetic Data**: The model is trained on synthetic data that simulates small motions, allowing it to learn how to accurately detect and magnify subtle movements.
- **Temporal Filtering**: Temporal filters are applied to the spatial representations to isolate motions within specific frequency bands. This allows the system to focus on small, subtle movements that occur within a certain frequency range, while ignoring larger, more pronounced movements that fall outside this range.
- **Learned Representations**:The network learns to extract features that are sensitive to small displacements. These learned representations are designed to be linear with respect to displacement, making them compatible with temporal filters. This linearity ensures that small movements are accurately captured and magnified.