Spot The Differences Between Two Images

Le Gia Khang
Le Duy Khang
Nguyen Hoang Tan

CS231: Introduction to Computer Vision

December 3, 2023

Presentation Overview

Presentation Overview

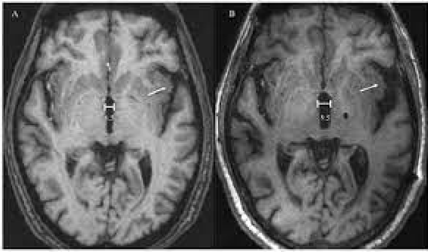
Why it Matters

Applications

Detecting changes is a natural computer vision task:

- The "spot-the-difference" game
- Facility monitoring
- Medical imaging
- Satellite surveillance
- Counterfeit detection

... and many more.



Comparison of two MRI scans over 10-year period.

Input and Output Analysis

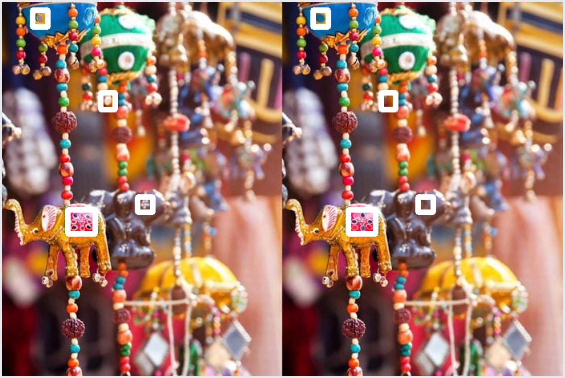
Input

Pair of images need to compare

Input and Output Analysis

Output

The provided images are marked with bounding boxes to indicate areas of distinction

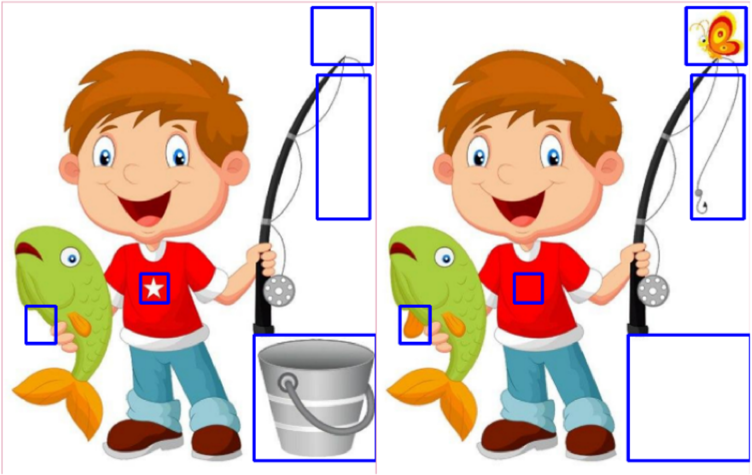


Examples: Input

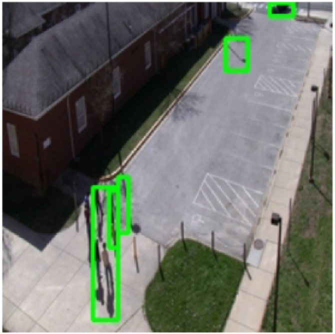
Examples: Input

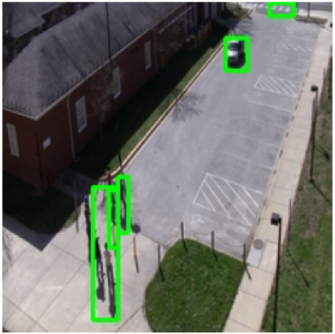
Examples: Input

Examples: Output

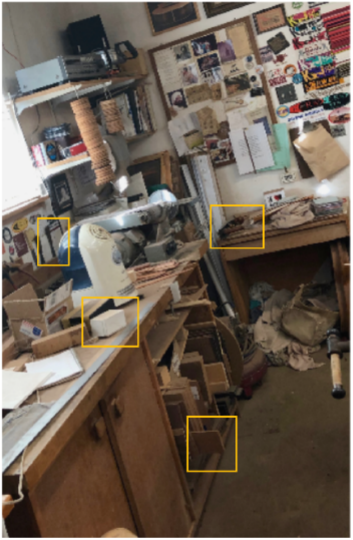


Examples: Output





Examples: Output



Presentation Overview

Presentation Overview

Pixel-Wise

Pixel-Wise Comparison

Compare the input image pair by first detecting which **pixels** have **changed** between the first and second image and then **segmenting** those pixels into **clusters** that *approximately* represent the objects that have changed.

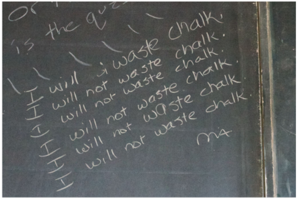
Or, in short:

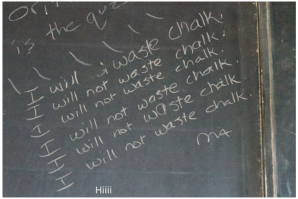
1. Detect pixels that have changed.
2. Perform clustering on those pixels.

Pixel-Wise: Detect changes in pixels

Base approach: pixel changes when its value is not the same.

Well, it works fine. . . with digital changes (e.g. photoshop,...)





Pixel-Wise: Detect changes in pixels

But fails horribly with non-digital changes (e.g. camera's noises, environmental changes, . . .)

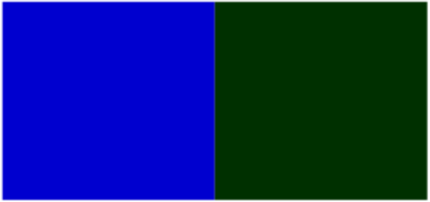
Pixel-Wise: Detect changes in pixels

So, **some changes are tolerable**. Hence, we will need to compute the **distance** between two pixels and compare that against a **threshold** value.

The most popular method of computing that distance is to compute the difference between the two pixels' **luminosity** (convert images into grayscale and then compare its values).

Pixel-Wise: Detect changes in pixels

However, some colours have similar **grayscale representation** despite not being the same color.



RGB(0, 0, 255) and RGB(0, 48, 0)

Convert to **grayscale**

Pixel-Wise: Detect changes in pixels

Distance between colours C_1 and C_2:

$$\bar{r} = \frac{C_{1,R} + C_{2,R}}{2}$$

$$\Delta R = C_{1,R} - C_{2,R}$$

$$\Delta G = C_{1,G} - C_{2,G}$$

$$\Delta B = C_{1,B} - C_{2,B}$$

$$\Delta C = \sqrt{\left(2 + \frac{\bar{r}}{256}\right) \times \Delta R^2 + 4 \times \Delta G^2 + \left(2 + \frac{255 - \bar{r}}{256}\right) \times \Delta B^2}$$

Pixel-Wise: Detect changes in pixels

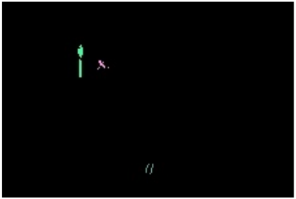
Figure: Result after applying the colour distance function

As we only care about **objects** that have changed not changed pixels, we have to group those pixels **into clusters** that approximate those objects.

And, due to the characteristics of those pixels (denser area is more likely to be an object than noises) **density based clustering algorithms** like *DBScan* will perform well.

Pixelwise: Clustering

Finally. encapsulate those clusters with bounding boxes.

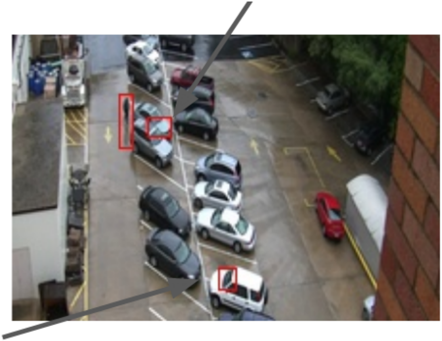


Pixel-wise optional: Remove non-interesting clusters

In images from a surveillance video, a lot of unnecessary changes are detected like slight movement, reflective area, . . .

These clusters can be reduced by using a simple scoring system:

$$\mathit{number\_of\_changed\_pixels} + \mathit{height\_of\_cluster} \times 2 + \mathit{width\_of\_cluster} \geq \mathit{threshold}$$



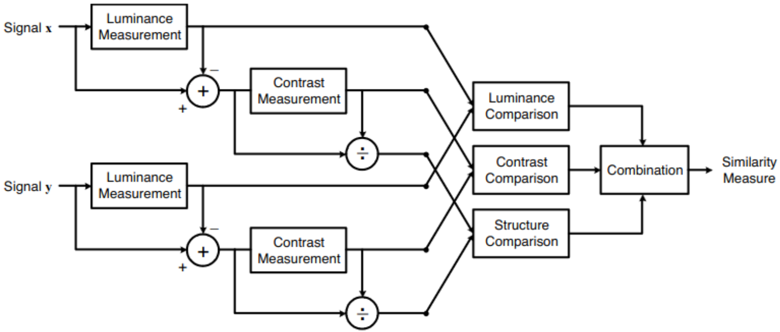
Presentation Overview

Structural similarity index (SSIM)

Structural similarity index (SSIM)

SSIM is used as a metric to measure the similarity between two given images which is a value between -1 and +1.

- Designed to mimic human perception in evaluating image quality.
- It takes into account *luminance*, *contrast*, and *structure*.



SSIM: Luminance

Luminance is measured by **averaging** over all the pixel values. Its denoted by μ and the formula is given below:

$$\mu_x = \frac{1}{N} \sum_{i=1}^{N} x_i$$

The luminance comparison function $\ell(x, y)$

$$\ell(x, y) = \frac{2\mu_{\mathbf{x}}\mu_{\mathbf{y}} + C_1}{\mu_{\mathbf{x}}^2 + \mu_{\mathbf{y}}^2 + C_1}$$

- $\mu_{\mathbf{x}}, \ \mu_{\mathbf{y}}$: Mean luminance of the first and second image.
- $C_1 = (K_1 L)^2$: where L = 255 for 8-bit component images.
- $K_1 \ll 1$: Default value is .01

SSIM: Contrast

Contrast is measured by taking the standard deviation (square root of variance) of all the pixel values. It is denoted by σ

$$\sigma_x = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu_x)^2}$$

The luminance comparison function (x, y)

$$c(x,y) = \frac{2\sigma_{\mathbf{x}}\sigma_{\mathbf{y}} + C_2}{\sigma_{\mathbf{x}}^2 + \sigma_{\mathbf{y}}^2 + C_2}$$

- $\sigma_{\mathbf{x}},\ \sigma_{\mathbf{y}}$: The standard deviation of pixel values
- $C_2 = (K_2 L)^2$: where L = 255 for 8-bit component images.
- $K_2 \ll 1$: Default value is .03

SSIM: Structure

The formula for covariance σ_{xy} between x and y is given by:

$$\sigma_{xy} = \frac{1}{N-1} \sum_{i=1}^{N} (x_i - \mu_x)(y_i - \mu_y)$$

Structure comparison function $s(x, y)$

$$s(x, y) = \frac{\sigma_{\mathbf{xy}} + C_3}{\sigma_{\mathbf{x}}\sigma_{\mathbf{y}} + C_3}$$

- $\sigma_{\mathbf{xy}}$: The covariance between x and y.
- $C_3 = C_2/2$.

The SSIM score

And finally, the SSIM score is given by,

$$\mathrm{SSIM}(\mathbf{x}, \mathbf{y}) = [l(\mathbf{x}, \mathbf{y})]^{\alpha} \cdot [c(\mathbf{x}, \mathbf{y})]^{\beta} \cdot [s(\mathbf{x}, \mathbf{y})]^{\gamma}$$

Where $\alpha > 0, \beta > 0, \gamma > 0$ denote the importance of each of the metrics.

To simplify the expression, if we assume, $\alpha = \beta = \gamma = 1$, we can get,

$$\mathrm{SSIM}(\mathbf{x}, \mathbf{y}) = \frac{(2\mu_x \mu_y + C_1)(2\sigma_{xy} + C_2)}{\left(\mu_x^2 + \mu_y^2 + C_1\right)\left(\sigma_x^2 + \sigma_y^2 + C_2\right)}.$$

Mean Structural Similarity Index

Instead of applying the above metrics globally, it's better to apply the metrics regionally

Mean Structural Similarity Index (MSSIM)

For image quality assessment, it is useful to apply the SSIM index locally rather than globally.

- Image statistical features are usually highly spatially nonstationary.
- Image distortions, which may or may not depend on the local image statistics, may also be space-variant.
- Only a local area in the image can be perceived with high resolution by the human observer at one time instance

Mean Structural Similarity Index

Gaussian Weighing

We use an 11x11 circular-symmetric Gaussian Weighing function

- Moves pixel-by-pixel over the entire image.
- At each step, the local statistics and SSIM index are calculated within the local window.

$$\mu_x = \sum_{i=1}^{N} w_i x_i$$

$$\sigma_x = \left(\sum_{i=1}^{N} w_i (x_i - \mu_x)^2 \right)^{\frac{1}{2}}$$

$$\sigma_{xy} = \sum_{i=1}^{N} w_i (x_i - \mu_x)(y_i - \mu_y).$$

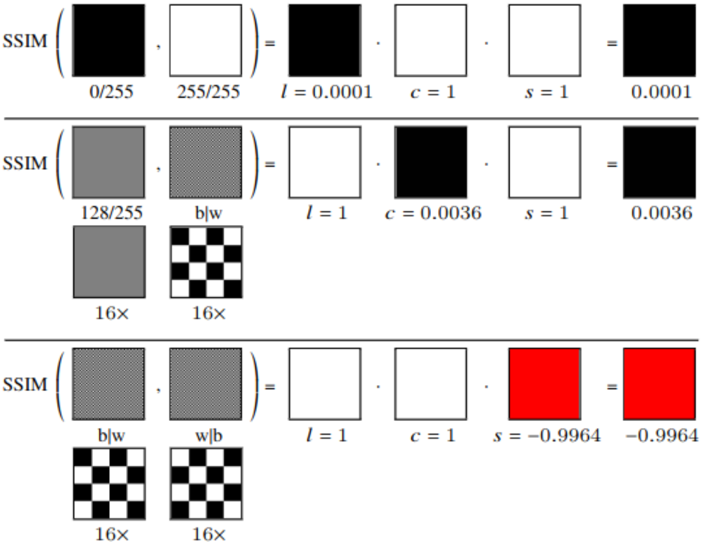
Where w_i is the gaussian weighting function.

Mean Structural Similarity Index

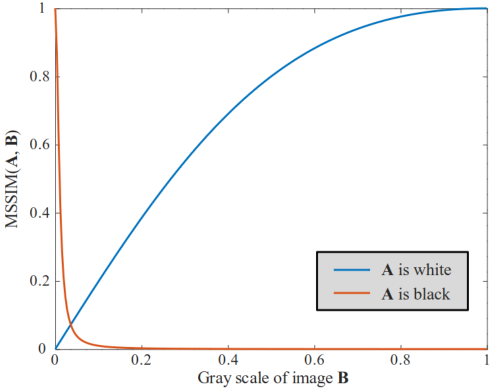
Once computations are performed all over the image, we simply take the *mean of all the local SSIM values* and arrive at the global SSIM value.

$$\mathrm{MSSIM}(\mathbf{X}, \mathbf{Y}) = \frac{1}{M} \sum_{j=1}^{M} \mathrm{SSIM}(\mathbf{x}_j, \mathbf{y}_j)$$

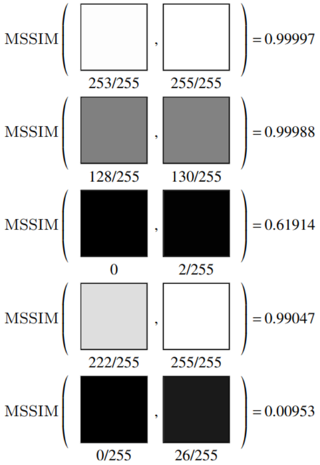
SSIM: Examples



SSIM: Examples



MSSIM as a function of an increasingly brighter, constant-valued image, B, against a black image (orange curve) and a white image (blue curve).



MSSIM behavior for constant grayscale images.

Presentation Overview

Siamese Network

Proposed in the article "The Change You Want To See" accepted at the IEEE/CVF Winter Conference on Application of Computer Vision 2023.

Ragav Sachdeva

Andrew Zisserman

Visual Geometry Group at the University of Oxford

Siamese Network: Overview

- Enable "object-level" change prediction and simplify counting the number of changes between two images.
- Use an architecture that operates on two images with geometric (scale, rotation,..) and photometric changes.
- Designed to be class-agnostic, it can detect changes irrespective of the object classes involved.

Model Architecture Overview

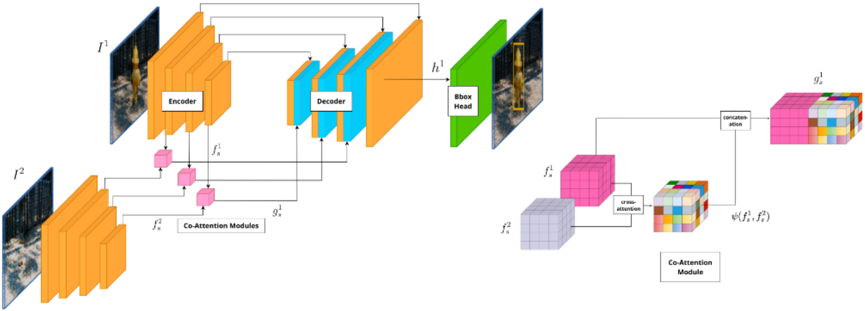


Figure: **Architecture**: Utilizing a dual-image encoder, feature maps (f_1^s, f_2^s) are generated. A co-attention module aligns and conditions these maps (g_1^s, g_2^s). Subsequently, a U-Net decoder processes the original and conditioned maps to yield final feature maps (h_1, h_2). The bounding box detector head employs h_1 and h_2 to generate bounding boxes for images I_1 and I_2, respectively

Siamese Network: Overview

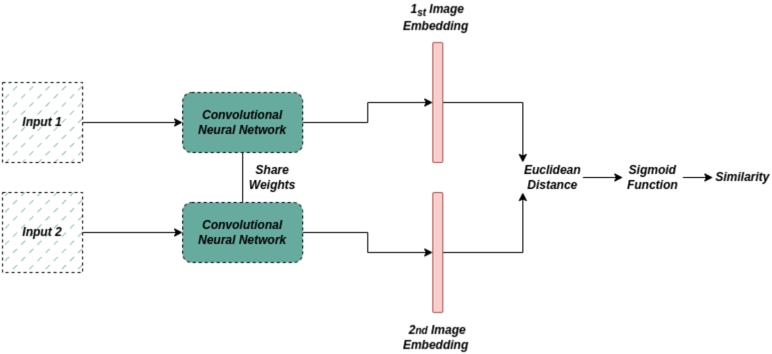
Siamese Network

A type of neural network architecture designed for tasks involving similarity or distance measurement between input pairs.

- Consists of two identical subnetworks (or twins) that share the same set of weights and parameters.
- The name originates from Chang (left) and Eng Bunker (right).

Siamese Network: Architecture

- Consists of two identical subnetworks.
- Extract feature vectors from both networks using a common set of convolutional and fully connected layers.
- Feature vectors from both networks are compared using a loss function L.

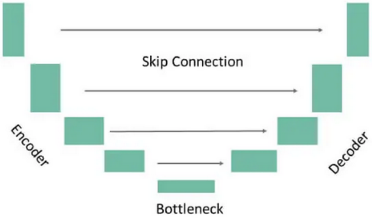


U-Net Encoder-Decoder Network

U-Net is a type of convolutional neural network (CNN) architecture commonly used for image segmentation tasks.

Consists of an encoder-decoder structure:

- **Encoder:** capturing features from the input image.
- **Decoder:** upsampling and producing a segmented output.



The authors employed **ResNet50** as the CNN for the encoder.

A UNet encoder-decoder with CoAM

CoAM Attention Module

We wish to concatenate features from both images in order to condition the model on both input images.

- However, for a given spatial location, the relevant feature in the other image may not be at the same spatial location.

As a result, we use an attention mechanism to model long range dependencies.

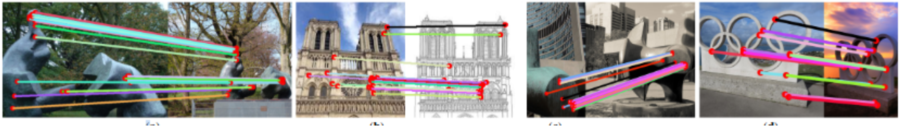
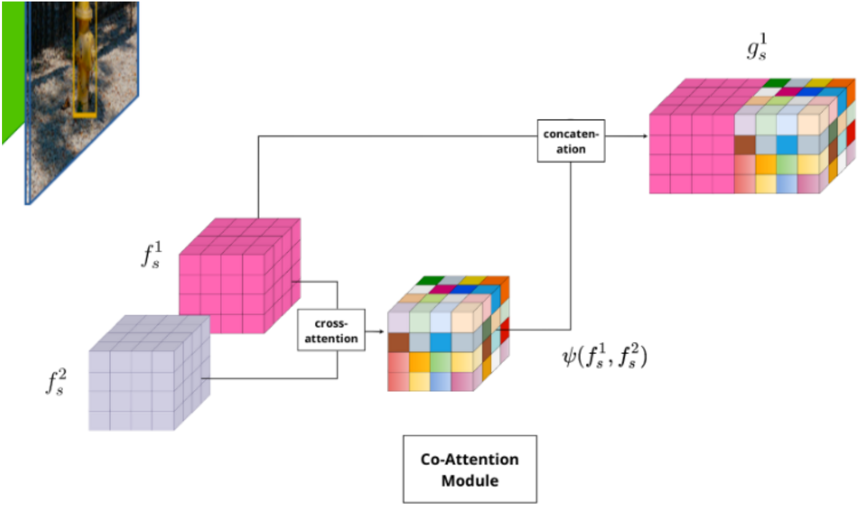
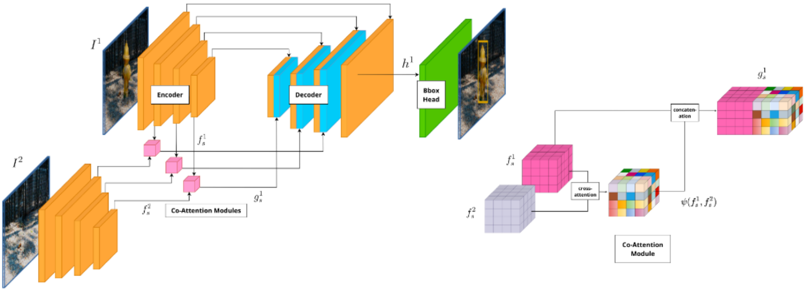


Figure: Correspondences obtained with the CoAM model, which is augmented with an attention mechanism.

A UNet encoder-decoder with CoAM

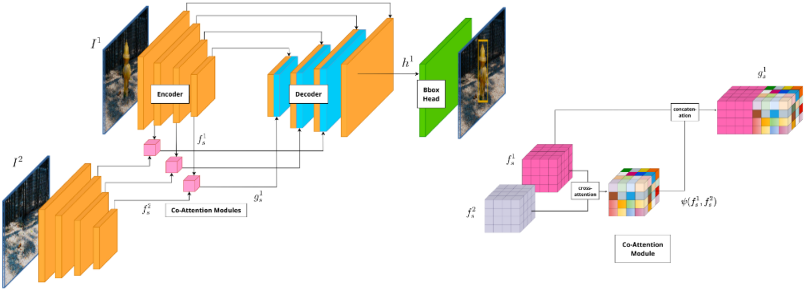


Use Siamese network to detect changes



- First obtaining a set of dense feature descriptors for each image using a CNN-based (ResNet50) encoder.
- These feature are then conditioned on each other using a co-attention mechanism that implicitly supplies the correspondences.

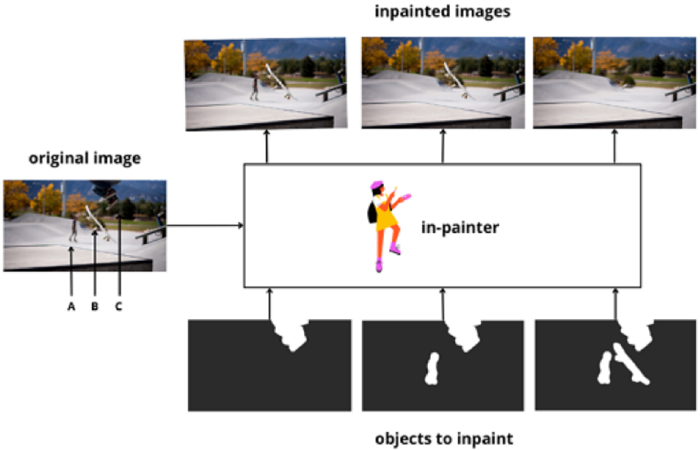
Use Siamese network to detect changes



- Next, feature are passed through a decoder to obtain high resolution conditioned image descriptors which are used by a bounding box detection head to localise the changes.

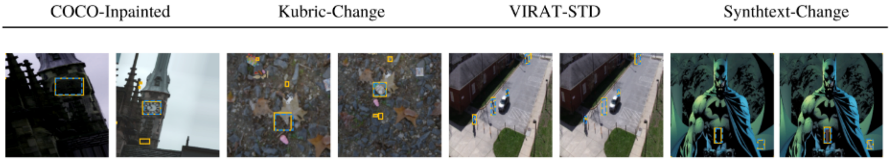
Siamese Network: Dataset

For this method, we make use of a state-of-the-art image inpainting method, **LaMa**, to make the objects *disappear*.



Siamese Network: Dataset

- We also apply random affine transformations to the images along with colour jittering or add random text to “background” images.
- Datasets: COCO-Inpainted, Synthtext-Change, VIRAT-STD, Kubric-Change.



Presentation Overview

VIRAT Video Dataset

VIRAT Video Dataset

A large-scale surveillance video dataset includes videos collected from stationary ground cameras and aerial vehicles. It has several event types:

- **Single Person Events** (8): walking, running, standing, throwing, gesturing, carrying, loitering, picking up
- **Person and Vehicle Events** (7): getting into or out of vehicle, opening or closing trunk, loading, unloading, dropping off, bicycling
- **Person and Facility Events** (2): entering or exiting facility



Evaluation Metrics

To assess the effectiveness of the proposed methods, we will employ key performance metrics: *precision, recall, and F1 score*.

We calculate these metrics for a set of ground truth and predicted bounding boxes corresponding to specific image.

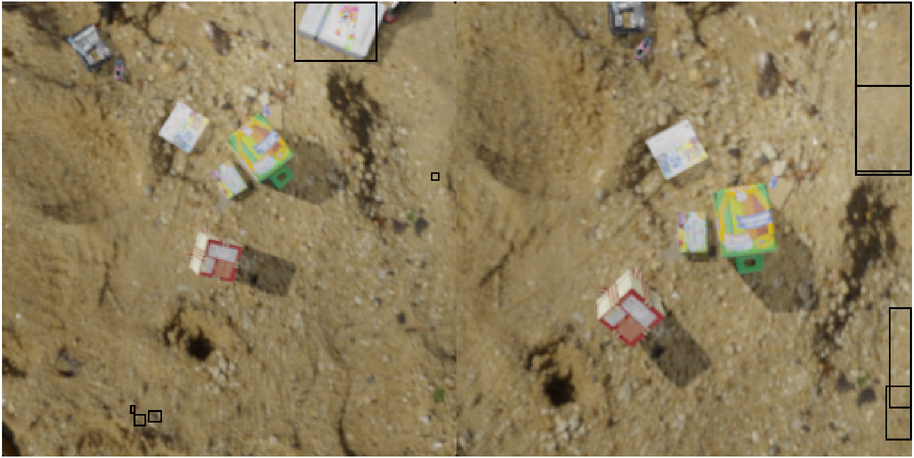
Table: Performance Evaluation of Different Approaches

Approach	**Precision**	**Recall**	**F1 Score**
Pixel-wise	.076	. 063	.069
SSIM	.314	.169	.22
Siamese Network	.354	.584	.441

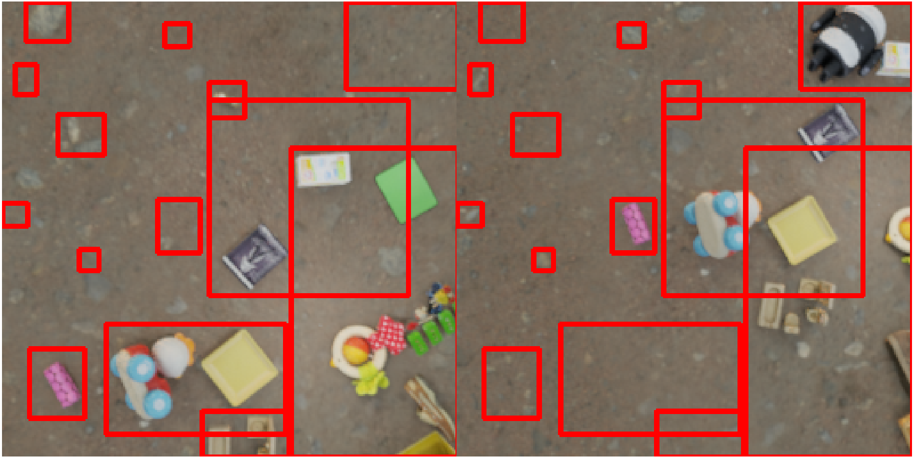
Output Examples: Siamese Network

Output Examples: Siamese Network

Output Examples: Siamese Network



Output Examples: SSIM



Output Examples: Pixel-wise

Reference

The Change You Want To See
Ragav Sachdeva and Andrew Zisserman

Image Quality Assessment: From Error Visibility to Structural Similarity
Zhou Wang, Alan Conrad Bovik, Hamid Rahim Sheikh and Eero P. Simoncelli

Datasets: The Change You Want to See (WACV 2023).
Ragav Sachdeva and Andrew Zisserman

Thanks for listening!

Q&A section