**A**

**Report**

**for Summer Research Internship**

**on**

**Video Based Attendance System**

**Submitted by**

**PRIYANSH PATHAK**

**JULY 2025**



**SUPERVISOR**

**DR. DINESH SINGH**

**(ASSISTANT PROFESSOR)**

**PHD. NANDANI SHARMA**

**SCHOOL OF COMPUTING AND ELECTRICAL ENGINEERING,**

**INDIAN INSTITUTE OF TECHNOLOGY MANDI AT KAMAND,**

**HIMACHAL PRADESH – 175075**

**TABLE OF CONTENTS**

|  |  |
| --- | --- |
| **CONTENTS** | **PAGE NUMBER** |
| Acknowledgement | i |
| Abstract | ii |
| Chapter 1: Introduction | 1 - 2 |
| Chapter 2: Relative Work | 3 - 5 |
| Chapter 3: Methodology | 6 -14 |
| 3.1 Overview | 6 - 10 |
| 3.2 Video Frame Sampling and Preprocessing | 10 - 15 |
| 3.2.1. Frame Sampling Strategy | 11 |
| 3.2.2. Image Preprocessing | 12 |
| 3.3 Face Detection and Alignment | 15 -19 |
| 3.3.1. Face Detection Using MTCNN | 15 |
| 3.3.2. Face Alignment | 15 |
| 3.4 Siamese Neural Network for Face Embedding  3.4.1 Model Architecture  3.4.2 Training with Triplet Loss  3.4.3 Hard Triplet Mining  3.5 Face Embedding Matching and AttendanceAssignment  3.5.1 Embedding Comparison  3.5.2 Distance Metric and Thresholding  3.5.3 Attendance Logging Logic  3.6 System Integration and Deployment  3.7 Summary of Mathematical Foundations | 18 |
| Chapter 4: Experiment Setup | 20 -24 |
| 4.1 Change-Aware Sampling and Contrastive  Learning for Satellite Images | 20 |
| 4.1.1. Datasets | 20 |
| 4.1.2. Implementation details and Baselines | 20 |
| 4.2 Fully Transformer Network for Change Detection of  Remote Sensing Images | 20-22 |
| 4.2.1. Datasets | 21 |
| 4.2.2. Evaluation Metrics | 21 |
| 4.2.3. Implementation Details | 22 |

|  |  |
| --- | --- |
| 4.3. DDPM-CD: Denoising Diffusion Probabilistic  Models as Feature Extractors for Change Detection | 22-24 |
| 4.3.1. Pre-training DDPM | 22 |
| 4.3.1. Pre-training DDPM | 22 |
| 4.3.2.1 Datasets | 23 |
| 4.3.3. Performance metrics | 24 |
| Chapter 5: Results | 25-26 |
| 5.1 .Change-Aware Sampling and Contrastive  Learning for Satellite Images | 25 |
| 5.1.1 Quantitative Results | 25 |
| 5.2. Fully Transformer Network for Change  Detection of Remote Sensing Images | 27-28 |
| 5.2.1. Comparisons with Sate-of-the-arts | 27 |
| 5.2.2. Quantitative Comparisons | 28 |
| 5.3. DDPM-CD: Denoising Diffusion Probabilistic  Models as Feature Extractors for Change Detection | 29-31 |
| 5.3.1. Quantitative change detection results | 29 |
| 5.3.2. Qualitative change detection results | 30 |
| Conclusion | 32 |
| Future Scope | 33 |
| References | 35 |

**ACKNOWLEDGEMENT**

The successful realization of the project is an outgrowth of a consolidated effort of the people from desperate fronts. It’s only with their support and guidance that the developer could meet the end. I would like to express my gratitude to IIT MANDI for providing me with the opportunity to undergo one month of training and for allowing me to delve into the intricacies of projects. I am sincerely thankful to **Dr. Dinesh Singh** (Assistant Professor) for guiding me throughout my training period. I also extend my appreciation to **PhD. Nandani Sharma**and other PhD students at IIT Mandi for their valuable guidance and support during this training.

**ABSTRACT**

In recent years, the need for automated and contactless attendance systems has significantly increased, especially in educational and corporate environments. Traditional methods of attendance marking, such as manual roll calls or biometric systems, are either time-consuming, prone to manipulation, or require physical contact.

To address these limitations, this project presents an **AI-powered video-based attendance system** that leverages deep learning and face recognition technologies to identify individuals from recorded classroom footage and mark their attendance automatically.

The system utilizes a **Siamese Neural Network architecture with triplet loss** to accurately learn facial embeddings that can distinguish between different individuals, even under variations in lighting, pose, and expression. The input video is processed frame-by-frame using **OpenCV**, and faces are detected, aligned, and matched against a database of pre-registered students. Once a match is found, the system records the individual's presence in a **CSV-based attendance log**.

To enhance recognition reliability, techniques such as **MTCNN** for face alignment and **feature distance thresholding** are employed. The backend is developed using **Flask**, allowing for future deployment as a web-based application. The system is scalable, efficient, and capable of operating offline, making it suitable for real-world academic settings. This project demonstrates the effectiveness of combining **deep learning**, **computer vision**, and **video processing** to create an intelligent and automated attendance solution.

**CHAPTER 1. INTRODUCTION**

With the rapid advancement in artificial intelligence and computer vision, traditional manual tasks are increasingly being automated to enhance efficiency and accuracy. One such area that has witnessed significant interest is **automated attendance management**, particularly in academic institutions and corporate settings where large-scale monitoring of individuals is required. Conventionally, attendance is recorded manually by instructors or through biometric systems like fingerprint scanners or ID card swipes. While these methods have served their purpose, they are not without drawbacks. Manual systems are time-consuming and prone to human error, while biometric systems require physical contact, which may not be suitable in terms of hygiene and efficiency, especially in the post-COVID-19 world.

In light of these challenges, there is a growing demand for **contactless, automated, and intelligent attendance systems**. This project aims to address that need by developing an **AI-powered video-based attendance system** that leverages deep learning techniques for face recognition to identify students from classroom recordings and mark their attendance automatically. This solution minimizes human intervention, reduces administrative workload, and improves reliability over traditional methods.

The core of the proposed system is a **Siamese Neural Network (SNN)** trained using the **triplet loss function**. This architecture is particularly effective for face recognition tasks as it learns to distinguish between similar and dissimilar facial features by mapping images into an embedding space where the distance between embeddings reflects the similarity between faces. The system first processes classroom video recordings using **OpenCV**, extracting frames and detecting faces using techniques such as **MTCNN (Multi-task Cascaded Convolutional Neural Network)**. Each detected face is then pre-processed and passed through the trained model to obtain a feature vector. These vectors are compared with a database of known student embeddings, and if the distance is within a specified threshold, the student's identity is confirmed, and attendance is marked in a **CSV-based log**.

This project not only demonstrates the practical implementation of deep learning and face recognition but also showcases the potential of combining multiple AI disciplines—such as computer vision, neural network training, and video processing—for solving real-world problems. The system has been developed using **Python**, with frameworks like **TensorFlow/Keras** for model training, and **Flask** for backend support and future web integration. The modular nature of the system allows for easy enhancements such as **real-time processing from live camera feeds**, **cloud deployment**, or **mobile compatibility**.

The motivation behind this project stems from the increasing need for automation in educational administration and the real-world applicability of face recognition systems. By replacing manual and semi-automated attendance systems with a fully automated, video-driven solution, institutions can ensure higher accuracy, save time, and eliminate redundancy. Moreover, this system serves as a foundation for further research and innovation in the areas of **smart surveillance**, **video analytics**, and **AI-assisted monitoring**.

In summary, this project provides a modern, efficient, and scalable alternative to traditional attendance systems by harnessing the power of deep learning and computer vision. It exemplifies how advanced technologies can be applied in meaningful ways to transform conventional processes and improve day-to-day operations in educational and professional settings.

**CHAPTER 2. RELATIVE WORK**

The automation of attendance systems has been a subject of research and development for many years. Traditional methods, such as manual roll calls or biometric systems like fingerprint and RFID-based tracking, are still widely used but suffer from several limitations, including inefficiency, physical contact requirements, and susceptibility to fraud or proxy attendance. These drawbacks have motivated the development of more advanced, intelligent solutions that leverage artificial intelligence and computer vision.

In recent years, the application of deep learning, particularly in the area of face recognition, has gained significant attention. Early face recognition systems based on techniques like Eigenfaces, Fisher faces, and LBPH had limited success in real-world environments due to sensitivity to lighting, pose, and facial expression changes. The emergence of deep convolutional neural networks (CNNs) brought about a transformative shift. Notable models such as Face Net, Deep Face, and VGGFace have demonstrated remarkable accuracy in recognizing faces by mapping them into high-dimensional embedding spaces where facial similarities can be measured using simple distance metrics like Euclidean or cosine distance.

A particularly effective architecture for such tasks is the Siamese Neural Network. Designed to determine similarity between two inputs, it becomes powerful when paired with the triplet loss function. The triplet loss operates on three images—an anchor, a positive (same identity), and a negative (different identity)—and trains the network to minimize the distance between the anchor and positive while maximizing the distance from the negative. This approach significantly improves the model's ability to generalize to unseen identities, making it highly suitable for face recognition in dynamic conditions, such as in classroom videos.

Face detection is another critical component of face recognition pipelines. Traditional approaches such as Haar cascades or HOG-based detectors have now been replaced by more robust deep learning-based solutions. One such method is the Multi-task Cascaded Convolutional Neural Network (MTCNN), which not only detects faces but also provides facial landmarks for alignment. Proper alignment plays a vital role in ensuring that the face embeddings generated by the recognition model are consistent, leading to better identification accuracy.

While face recognition from static images is a well-studied problem, video-based face recognition introduces additional challenges such as motion blur, occlusions, varying camera angles, and background clutter. Only a limited number of existing systems are capable of processing video input to perform reliable face recognition and attendance marking. Some methods rely on real-time video streams, while others process recorded videos offline, but many still lack automation in converting recognition results to structured attendance records.

This project builds upon the strengths of these prior works, integrating Siamese networks, triplet loss, face detection with MTCNN, and frame-wise video analysis to create a fully automated, video-based attendance system. Unlike many previous efforts that treat recognition and logging as separate components, this system delivers an end-to-end solution that processes classroom footage, detects and recognizes student faces, and logs attendance without manual intervention. It thus bridges the gap between face recognition research and its real-world application in educational institutions.

**CHAPTER 3. METHODOLOGY**

**3.1 Overview**

The main goal of this project is to automatically mark attendance using videos recorded in classroom environments. The system recognizes student faces in the video, matches them against a pre-registered database, and logs attendance. To achieve this, the methodology combines computer vision techniques including face detection, alignment, deep feature extraction with a Siamese neural network, and efficient matching to handle large datasets.

The methodology is divided into the following key components:

* Video Frame Sampling and Preprocessing
* Face Detection and Alignment
* Siamese Network for Face Embedding
* Embedding Matching and Identity Assignment
* Attendance Logging
* System Integration and Deployment

The subsequent sections describe each step in detail.

**3.2 Video Frame Sampling and Preprocessing**

**3.2.1 Frame Sampling Strategy**

Processing every frame of a classroom video can be computationally expensive and redundant because many frames are visually similar. Therefore, we apply a frame sampling technique to reduce computation:

* Videos are sampled at a rate of 1 frame every 1 second (configurable).
* This rate is chosen to balance temporal coverage with resource constraints.

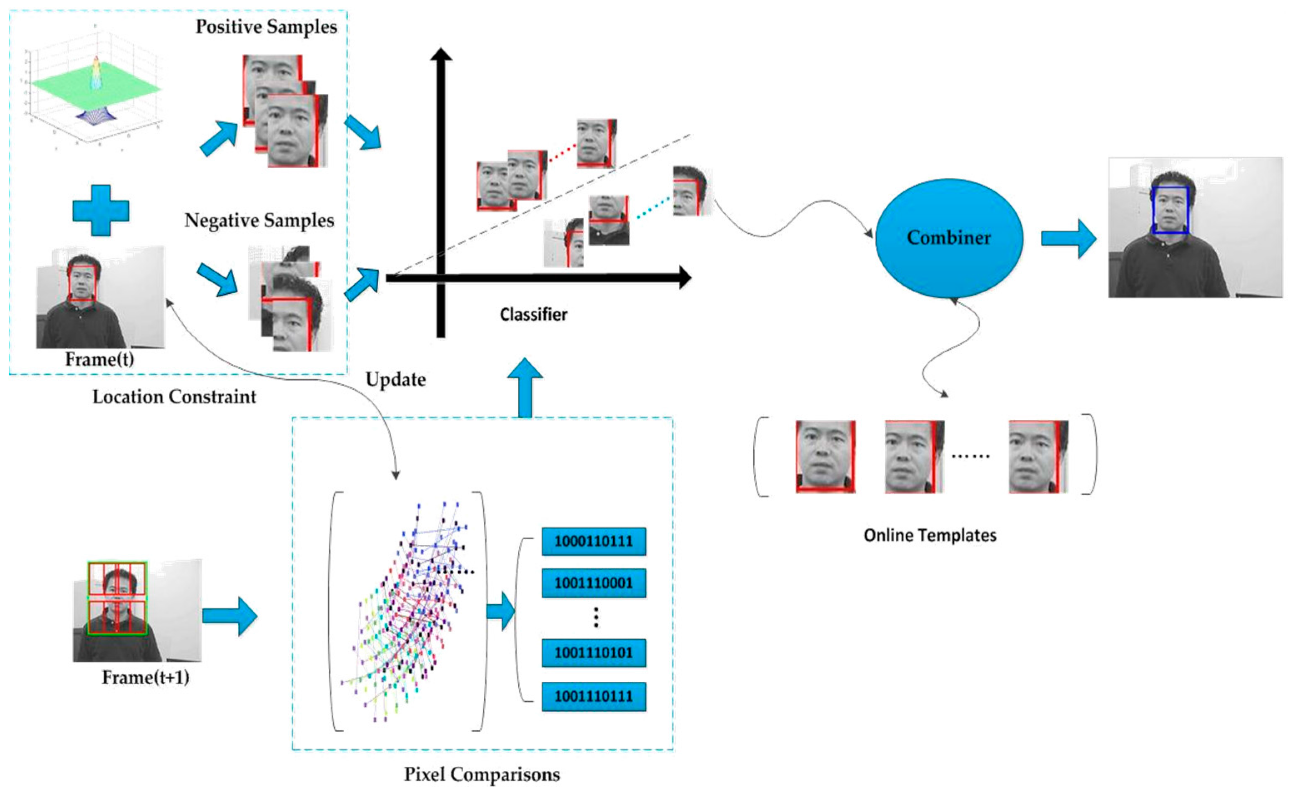
By sampling one frame per second, the system ensures sufficient coverage of the class while keeping processing manageable.

**3.2.2 Image Preprocessing**

Each sampled frame undergoes preprocessing:

* Colour normalization: Convert images to RGB and normalize pixel intensities to [0,1].
* Resizing: Standardize frame size to 720p (1280×720) to maintain uniformity across videos.

This standardization improves model performance and makes subsequent face detection more consistent.

Figure 3.1: Frame sampling and preprocessing flowchart.

**3.3 Face Detection and Alignment**

**3.3.1 Face Detection Using MTCNN**

Faces within each sampled frame are detected using the Multi-task Cascaded Convolutional Network (MTCNN), which has three stages:

1. Proposal Network (P-Net): Generates candidate face windows with bounding boxes.
2. Refine Network (R-Net): Refines candidate boxes by removing false positives.
3. Output Network (O-Net): Outputs final bounding boxes with facial landmarks (eyes, nose, mouth corners).

MTCNN is chosen for its balance between speed and accuracy, and its ability to detect multiple faces per frame.

**3.3.2 Face Alignment**

Detected faces are aligned to a canonical pose using the 5 facial landmarks:

* Left eye, right eye, nose, left mouth corner, right mouth corner.
* A similarity transform (rotation, scaling, translation) warps the face so that the eyes and mouth align with fixed reference points.

This normalization reduces pose variation, which significantly improves recognition accuracy.

The aligned face is then cropped to a fixed size of 160×160 pixels.

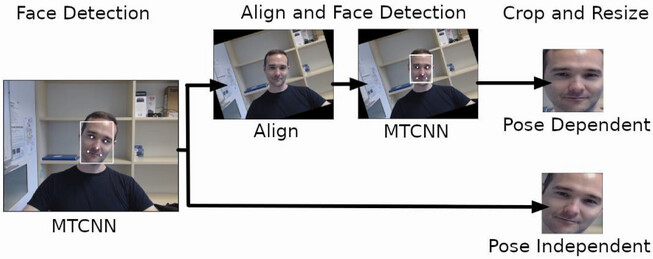


Figure 3.2: Face detection and alignment example.

**3.4** **Siamese Neural Network for Face Embedding**

**3.4.1 Model Architecture**

The core of the recognition system is a Siamese Neural Network trained to produce a 128-dimensional embedding vector for each aligned face image. The network uses a convolutional backbone similar to Face Net, consisting of:

* Multiple convolutional layers with batch normalization and ReLU activations.
* Max-pooling layers to down sample feature maps.
* Two fully connected layers projecting to a compact embedding space.

This architecture maps each face image xxx to an embedding vector:

**f(x) ∈ ℝ¹²⁸**

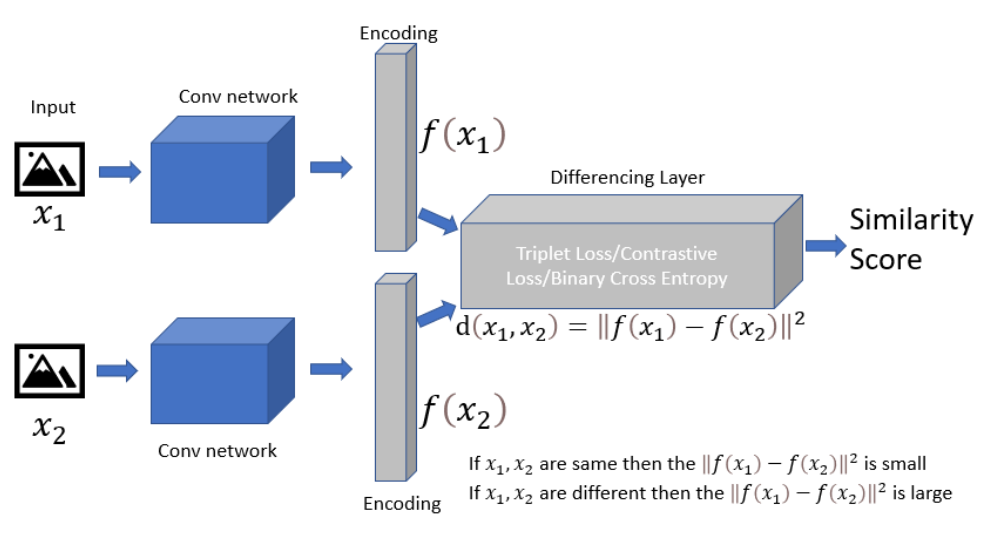


Figure 3.3: Siamese Neural Network for Face Embedding

**3.4.2 Training with Triplet Loss**

The Siamese network is trained using triplet loss, which encourages embeddings of the same person (anchor a and positive p) to be close and embeddings of different persons (negative n) to be far apart.

The loss for each triplet (xa​, xp​,xn​) is defined as:

**LTriplet = max(||f(xₐ) − f(xₚ)||² − ||f(xₐ) − f(xₙ)||² + α, 0)**

where:

* ||.||2​ is the Euclidean norm.
* α is a margin parameter (set to 0.2) that enforces a minimum separation between positive and negative pairs.

3.4.3 Hard Triplet Mining

To improve training efficiency and convergence:

* For each anchor, the hardest positive and hardest negative examples within a mini-batch are selected.
* Hard positives are the same person images with largest distance to anchor.
* Hard negatives are different person images with smallest distance to anchor.

This mining strategy focuses the model on difficult examples.

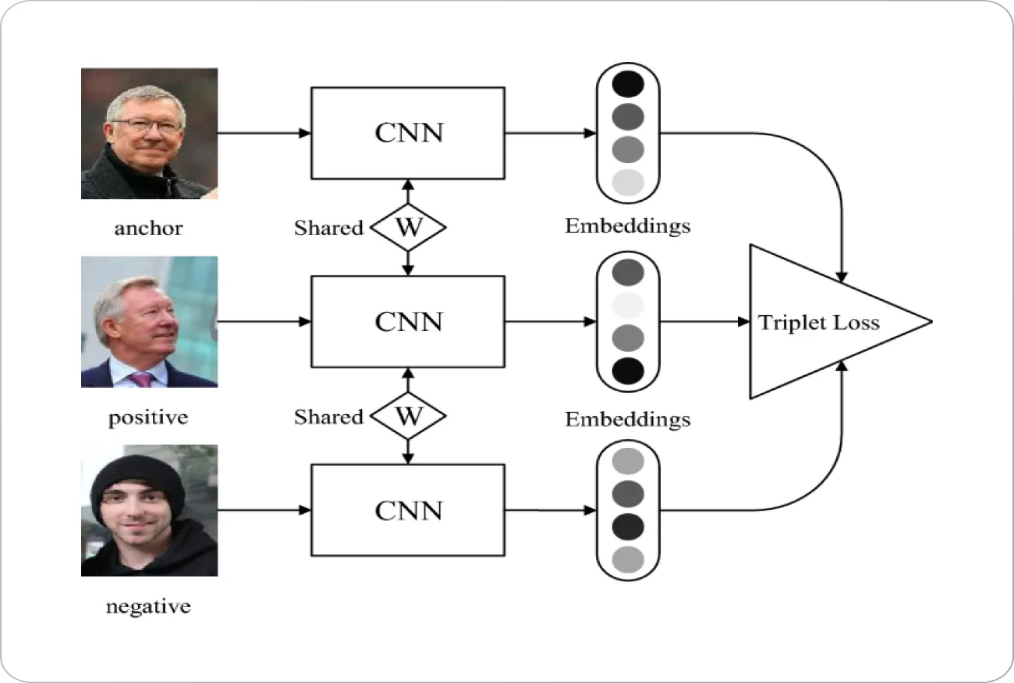


Figure 3.3: Siamese network training with triplet loss.

**3.5 Face Embedding Matching and Attendance Assignment**

**3.5.1 Embedding Comparison**

For each detected and aligned face in the video frame during inference:

* Compute the embedding e = f(x).
* Compare e with embeddings of registered students {Ei}.

3.5.2 Distance Metric and Thresholding

The Euclidean distance di between embeddings is calculated as:

**di=∥e−Ei∥2**

The student is identified as the one with minimum distance:

**i^= arg minidi**

If di^<τ (threshold τ set via validation, typically ~0.8), the face is assigned to student I; otherwise, it is considered unknown.

3.5.3 Attendance Logging Logic

To avoid multiple logging for the same student in one session:

* Maintain an in-memory attendance list per video session.
* Once a student is recognized and logged, ignore subsequent detections.

Each attendance entry records:

StudentID, Name, Timestamp, VideoSessionID

This is saved in CSV format for further processing.

**3.6 System Integration and Deployment**

The entire pipeline is implemented in Python, using:

* OpenCV for video handling and frame extraction.
* TensorFlow/Keras for running MTCNN and the Siamese network.
* Pandas for attendance data handling.
* A modular design allows easy swapping of components, such as changing the embedding model or adding web interface.

A web interface using Flask allows users to upload videos and download attendance reports. The system supports batch processing for multiple videos.

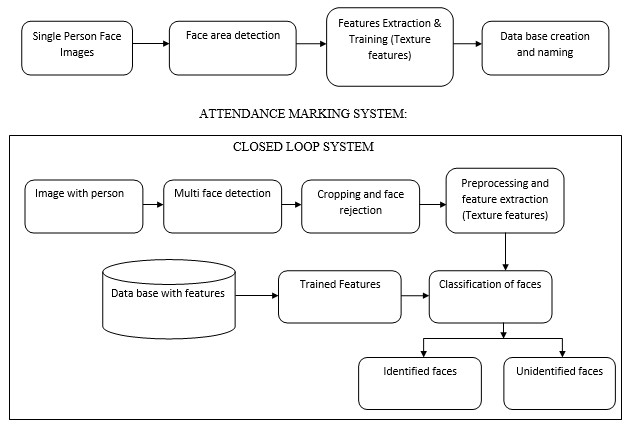


Figure 3.5: System architecture diagram.

| **Component** | **Mathematical Expression** | **Description** |
| --- | --- | --- |
| Face Alignment | T = argminₜ ∑⁵ᵢ₌₁ ‖T(pᵢ) − qᵢ‖² | Similarity transform for landmark alignment |
| Embedding Function | f: Image → ℝ¹²⁸ | CNN mapping images to embedding vectors |
| Triplet Loss | L = max(‖f(a) − f(p)‖² − ‖f(a) − f(n)‖² + α, 0) | Encourages anchor-positive closeness, anchor-negative distance |
| Distance Metric | dᵢ = ‖e − Eᵢ‖₂ | Euclidean distance between embeddings |
| Attendance Threshold | dᵢ < τ | Threshold for accepting identity match |

**3.7 Summary of Mathematical Foundations**

**CHAPTER 4: EXPERIMENT SETUP**

**4.1 Change-Aware Sampling and Contrastive Learning for Satellite**

**Images**

**4.1.1. Datasets**

Datasets with 100k images and the other with 1 million images were used for pre training of the model. These datasets contain RGB images from Sentinel-2 . To achieve the aim to learn a representation without supervision, which leads to better performance on all these downstream tasks. The image data for these algorithms and datasets are obtained from a few different sources. Sentinel-2 provides multispectral images at 10m resolution with a temporal revisit of 5 days. I use Sentinel-2 satellite imagery as it provides frequent temporal information that we use in the self-supervised formulation. The Sentinel-2 imagery consists of 12 spectral bands (including RGB and NIR) at 10 m, 20 m and 60 m resolution, with a revisit time of around 5 days. with use of Google Earth Engine to process and download image patches from about 200K locations around the world, where each patch covers a region of roughly 2.65×2.65 km datasets where prepared. At each location, they download 5 images from different dates separated by approximately 3 months, which capture the seasonal changes that occurred in the region over a year. To avoid getting images from the same periods of the year, at each location we jitter the dates for up to a year. We also filter out Sentinel-2 tiles with a cloud percentage higher than 10%. In total, we obtain about 1 million multi-spectral image patches, which amount to a total of over 387 billion pixels. We do not perform any additional data cleaning to ensure that the obtained images are diverse, informative and free of clouds. Because our dataset is constructed automatically, we can easily gather more data. In this work, however, we limit the scale to a total of 1M images to make it more comparable to ImageNet .

**4.1.2. Implementation details and Baselines**

Models are trained with a batch size of 256 and with 16,384 negative embeddings. We use an SGD optimizer with a momentum of 0.9 and a weight decay of 1e-4. We set an initial learning rate of 0.03 and divide it by 10 at 60% and 80% of the epochs. A temperature scaling τ of 0.07 is used in the contrastive loss. We experiment with two datasets for self-supervised pretraining: one with 100k images and the other with 1 million images. These datasets contain RGB images from Sentinel-2 . The long-time difference between the two sets and is 4 years. The maximum time difference within a set is 1 year. For training the model baselines are kept 1000 and 200 epochs on the 100k and 1m dataset respectively.

**4.2 Fully Transformer Network for Change Detection of Remote**

**Sensing Images**

**4.2.1. Datasets**

* LEVIR-CD [13]

LEVIR-CD is a public large-scale CD dataset which contains 637 remote sensing image pairs of 1024×1024 resolution (0.5m).then I split, and crop original images into small patches of size 256×256 with no overlapping. Therefore, we obtain 7120/1024/2048 pairs of image patches for training/validation/test, respectively.

* WHU-CD [14]

WHU-CD is a public building CD dataset. It contains one pair of high-resolution (0.075m) aerial images of size 32507×15354. Cropped the original image into small patches of size 256×256 with no overlap and randomly split it into three parts: 6096/762/762 for training/validation/test, respectively.

* SYSU-CD [15]

SYSU-CD is a public building CD dataset. It contains 20000 pairs of high-resolution (0.5m) images of size 256×256. I split dataset for experiments. There are 12000/4000/4000 pairs of image patches for training/validation/test, respectively.

* Google-CD [16]

is a very recent and public CD dataset. It contains 19 image pairs, originating from Google Earth Map. The image resolutions are ranging from 1006×1168 pixels to 4936×5224 pixels. We crop the images into small patches of size 256×256 with no overlap and randomly split it into three parts: 2504/313/313 for training/validation/ test, respectively.

**4.2.2. Evaluation Metrics**

To verify the performance, I followed the same as the paper author and mainly utilize F1, & Intersection over Union (IoU) scores (with regard to the change-class as the primary evaluation metrics), precision and recall of the change category and overall accuracy (OA).

**4.2.3. Implementation Details**

I perform experiments with supercomputer PARAM Himalaya based on heterogeneous and hybrid configuration of Intel Xeon Cascade Lake processors, and NVIDIA Tesla V100 GPU cards. I used the mini-batch SGD algorithm to train our framework with an initial learning rate 10-3, moment 0.9 and weight decay 0.0005. The batch size is set to 6. For the Siamese feature extraction backbone, we adopt the Swin Transformer pre-trained on ImageNet-22k classification task [17]. To fit the input size of the pre-trained Swin Transformer, we uniformly resize image patches to 384×384. For other layers, we randomly initialize them and set the learning rate with 10 times than the initial learning rate. We train the framework with 100 epochs. The learning rate decreases to the 1/10 of the initial learning rate at every 20 epoch. Encoder depth is set to (2,4,6,8). To improve the robustness, data augmentation is performed by random rotation and flipping of the input images. For the loss function in the model training, the weight parameters of each level are set equally.

**4.3. DDPM-CD: Denoising Diffusion Probabilistic Models as Feature**

**Extractors for Change Detection**

**4.3.1. Pre-training DDPM**

We pre-train a state-of-the-art, pixel-space, unconditional U-Net-like diffusion model on remote sensing images collected from the Google Earth Engine without any human supervision. The pre-training is conducted on our DDPM using remote sensing images of resolution 256 × 256. Our DDPM model comprises of two convolutional residual blocks per resolution level and self-attention blocks at the 16 × 16 resolution, positioned between the convolutional blocks. Group normalization with a group size of 32 is employed for normalization in both residual blocks and self-attention blocks, while a dropout rate of 0.2 is utilized in residual blocks. All self-attention blocks consist of one attention head. Diffusion time t is specified by adding the sinusoidal position embedding into each residual block. Our U-Net encoder includes five spatial scales (resolution levels). We set the base channel dimension to 128 and define the channel multiplier as {1, 2, 4, 8, 8}. A cosine βt schedule is chosen, setting T = 2000 without a sweep, and implementing a linear schedule from β1 = 10−6 to βt = 0.01. I utilize the Adam optimizer with a linear warm-up schedule over 10, 000 training steps, followed by a fixed learning rate of 1×10−5. The batch size is set to 8, and no exponential moving average (EMA) is applied to model parameters.

**4.3.2. Fine-tuning for change detection**

**4.3.2.1 Datasets**

Fine-tuning for change detection was conducted on four publicly available datasets: LEVIR-CD [13], WHUCD [14], DSIFN-CD [18], and CDD [19].

* The LEVIR-CD [13] dataset comprises 637 pairs of high-resolution remote sensing images with a spatial size of 1024 × 1024 and a spatial resolution of 0.5m, collected from Google Earth. It provides a total of 31,333 change labels for building instances, covering both building appearances and disappearances. The dataset includes official splits for train, val, and test, with 44,564, 564, and 128 pairs, respectively. Each 1024×1024 image is pre-processed into non-overlapping 256 × 256 patches for train, val, and test sets.
* The WHU-CD [14] dataset consists of paired aerial images acquired in 2012 and 2016, covering an area of 20.5km2, containing 12,796 and 16,077 building instances, respectively. The spatial size of each image is 15, 354 × 32, 507 pixels, with a spatial resolution of 0.2m. The dataset represents an area affected by a 6.3-magnitude earthquake in February 2011, resulting in numerous changes, including rebuilt buildings and new constructions. Semantic labels for building changes are available. Each image was divided into non-overlapping 256× 256 patches for train, val, and test sets.
* The DSIFN-CD [18] dataset comprises of six bi-temporal high-resolution images from six cities in China, clipped into 394 sub-image pairs of sizes 512 × 512. After augmentation, it contains 3940 bi-temporal image pairs. The training set includes 3600 image pairs, the validation set includes 340 image pairs, and the test set includes 48 image pairs. Each image was pre-processed into non-overlapping 256×256 patches for train, val, and test sets.
* The CDD [19] dataset consists of season-varying remote sensing images obtained from Google Earth. It contains 7 pairs of season-varying images with a resolution of 4725 × 2700 pixels for manual ground truth creation and 4 pairs with minimal changes and a resolution of 1900×1000 pixels for additional manual object inclusion. The dataset features objects of varying sizes (e.g., cars to large construction structures) and seasonal changes in natural objects (e.g., from a single tree to a wide forest area). The final dataset contains 16000 image pairs of size 256 × 256: 10000 for training, 3000 for validation, and 3000 for testing.

For Fine-tuning we optimize the parameters of the hierarchical change decoder and the change classifier using the Binary Cross-Entropy (CE) loss and the AdamW optimizer. The parameters of the pre-trained DDPM remain frozen during the fine-tuning process. Our initial learning rate is set to 1×10−4, linearly decaying to zero over 120 epochs. The evaluation metrics are reported on the test-set.

**4.3.3. Performance metrics**

To measure the change detection performance, I have use F1, Intersection over Union (IoU) scores (with regards to the change class as the primary quantitative indices) and overall accuracy (OA) to get a global quality of change predictions.

**CHAPTER 5: RESULTS**

**5.1 .Change-Aware Sampling and Contrastive Learning for Satellite**

**Images**

**5.1.1 Quantitative Results**

author evaluate their method on the scene recognition task on Functional Map of the World (FMoW). The dataset contains satellite views of images at multiple times for 62 different scenes such as parks, airports, shipyards, etc. Since the dataset, is at a different resolution author downscale images to match with the resolution of Sentinel-2 imagery. Additionally, since the dataset is originally designed for a different task of object detection using multitemporal images, & only use a single image for each scene.

Author added a linear layer to the pre-trained backbone. Similar to EuroSat Evaluation they perform the optimization by minimizing the cross-entropy loss over 62 classes. they use an Adam optimizer with default hyperparameters with a learning rate of 10−3. they trained the classifier for 100 epochs and reduce the learning rate by a factor of 10 at epochs 60 and 80, while using a batch size of 32 for training. Table 1 shows the top-1 and top-5 classification accuracy of various pre-trained backbones on the FMoW dataset. Method results in better features that are better suited for scene recognition.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Data | Pre-training | ResNet-18 | | ResNet-50 | |
| top-1 | top-5 | top-1 | top-5 |
| - | Random init. | 16.04 | 16.04 | 16.04 | 16.04 |
| ImageNet. | 32.41 | 61.22 | 37.31 | 65.03 |
| 100k | MoCo v2 | 34.33 | 63.17 | 38.27 | 67.25 |
| SeCo | 34.57 | 63.12 | 38.32 | 66.68 |
| CACo (author) | 36 | 64.72 | 39.9 | 68.59 |
| 1m | SeCo | 38.84 | 67.35 | 43.64 | 71.89 |
| CACo (author) | 39.13 | 68.06 | 44.12 | 72.52 |

Table 1. Performance of authors representation on the Functional Map of the World (FMoW) scene recognition task with linear probing, in top-1 and top-5 Accuracy. their method provides a more accurate classification, with different backbones.

|  |  |  |  |
| --- | --- | --- | --- |
| Backbone | Data | Pre-training | Fine-tuning |
| ResNet-18 | - | Random init. | 80.08 |
| - | ImageNet | 92.08 |
| 100k | MoCo v2 | 94.94 |
| SeCo | 96.71 |
| CACo (author) | 97.02 |
| 1m | SeCo | 97.25 |
| CACo (author) | 97.47 |
| ResNet-50 | - | Random init. | 79.2 |
| ImageNet | 93.41 |
| 100k | SeCo | 96.56 |
| CACo (author) | 97.17 |
| 1m | SeCo | 97.34 |
| CACo (author) | 97.77 |

Table 2. Performance of method on the EuroSat landcover classification task when they finetune the whole network.

EuroSat finetuning. Table 2 shows the performance of method on EuroSat classification when they perform finetuning instead of linear classification. Even in the case of fine-tuning they see gains, albeit small. This shows that not only does method learned a good representation, but it can also be used as a better initialization for transfer learning.

Training the model on the 100k dataset is still in progress and the implementation of the paper is still not completed.

**5.2. Fully Transformer Network for Change Detection of Remote Sensing Images**

**5.2.1. Comparisons with Sate-of-the-arts**

In this section, the experimental results of the proposed method are compared with other outstanding methods and FTN paper on four public CD datasets. The experimental results showcase the effectiveness of self-trained model with the results to the author the proposed method .

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Methods | LEVIR-CD | | | | | WHU-CD | | | | |
| Pre. | Rec. | F1 | IoU | OA | Pre. | Rec. | F1 | IoU | OA |
| FC-EF | 86.9 | 80.2 | 83.4 | 71.5 | 98.4 | 71.6 | 67.3 | 69.4 | 53.1 | 97.6 |
| FC-Siam-Di | 89.5 | 83.3 | 86.3 | 75.9 | 98.7 | 47.3 | 77.7 | 58.8 | 41.7 | 95.6 |
| FC-Siam-Conc | 92 | 76.8 | 83.7 | 72 | 98.5 | 60.9 | 73.6 | 66.6 | 50 | 97 |
| BiDateNet | 85.7 | 90 | 87.8 | 78.2 | 98.5 | 78.3 | 71.6 | 74.8 | 59.7 | 81.9 |
| U-Net++MSOF | 90.3 | 81.8 | 85.9 | 75.2 | 98.4 | 92 | 89.4 | 90.7 | 82.9 | 97 |
| DTCDSCN | 88.5 | 86.8 | 87.7 | 78.1 | 98.8 | 63.9 | 82.3 | 72 | 56.2 | 97.4 |
| DASNet | 80.8 | 79.5 | 79.9 | 74.7 | 94.3 | 68.1 | 73 | 70.5 | 54.4 | 97.3 |
| STANet | 83.8 | 91 | 87.3 | 77.4 | 98.7 | 79.4 | 85.5 | 82.3 | 70 | 98.5 |
| MSTDSNet | 85.5 | 90.8 | 88.1 | 78.7 | 98.6 |  |  |  |  |  |
| IFNet | 94 | 82.9 | 88.1 | 78.8 | 98.9 | 96.9 | 73.2 | 83.4 | 71.5 | 98.8 |
| SNUNet | 89.2 | 87.2 | 88.2 | 78.8 | 98.8 | 85.6 | 81.5 | 83.5 | 71.7 | 98.7 |
| BIT | 89.2 | 89.4 | 89.3 | 80.7 | 98.9 | 86.6 | 81.5 | 84 | 72.4 | 98.8 |
| H-TransCD | 91.5 | 88.7 | 90.1 | 81.9 | 99 | 93.9 | 88.7 | 91.2 | 83.9 | 99.2 |
| ChangeFormer | 92.1 | 88.8 | 90.4 | 82.5 | 99 | 91.8 | 88 | 89.9 | 81.6 | 99.1 |
| FTN (paper) | 92.7 | 89.4 | 91 | 83.5 | 99.1 | 93.1 | 91.2 | 92.2 | 85.5 | 99.4 |
| ours | 90.2 | 82.7 | 86.3 | 75.9 | 98.7 | 93.5 | 89.8 | 91.6 | 84.5 | 99.3 |

Table 3. Quantitative comparisons on LEVIR-CD and WHU-CD datasets.

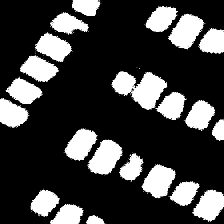
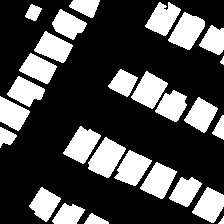
|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Methods | SYSU-CD | | | | | Google-CD | | | | |
| Pre. | Rec. | F1 | IoU | OA | Pre. | Rec. | F1 | IoU | OA |
| FC-EF | 74.3 | 75.8 | 75.1 | 60.1 | 86 | 80.8 | 64.4 | 71.7 | 55.9 | 85.9 |
| FC-Siam-Di | 89.1 | 61.2 | 72.6 | 57 | 82.1 | 85.4 | 63.3 | 72.7 | 57.1 | 87.3 |
| FC-Siam-Conc | 82.5 | 71 | 76.4 | 61.8 | 86.2 | 82.1 | 64.7 | 72.4 | 56.7 | 84.6 |
| BiDateNet | 81.8 | 72.6 | 76.9 | 62.5 | 89.7 | 78.3 | 71.6 | 74.8 | 59.7 | 81.9 |
| U-Net++MSOF | 81.4 | 75.4 | 78.3 | 62.1 | 86.4 | 91.2 | 57.6 | 70.6 | 54.6 | 95.2 |
| DASNet | 68.1 | 70 | 69.1 | 60.7 | 80.1 | 71 | 44.9 | 55 | 37.9 | 90.9 |
| *STANet* | 70.8 | 85.3 | 77.4 | 63.1 | 88 | 89.4 | 65 | 75.3 | 60.4 | 82.6 |
| DSAMNet | 74.8 | 81.9 | 78.2 | 64.2 | 89.2 | 72.1 | 80.4 | 76 | 61.3 | 94.9 |
| MSTDSNet | 79.9 | 80.8 | 80.3 | 67.1 | 90.7 | - | - | - | - | - |
| SRCDNet | 75.5 | 81.1 | 78.2 | 64.2 | 89.3 | 83.7 | 71.5 | 77.1 | 62.8 | 83.2 |
| BIT | 82.2 | 74.5 | 78.2 | 64.1 | 90.2 | 92 | 72 | 80.8 | 67.8 | 96.6 |
| H-TransCD | 83.1 | 77.4 | 80.1 | 66.8 | 91 | 85.9 | 81.7 | 83.8 | 72.1 | 97.6 |
| FTN (paper) | 86.9 | 76.8 | 81.5 | 68.8 | 91.8 | 87 | 84.2 | 85.6 | 74.8 | 97.9 |
| ours | 79.5 | 70.6 | 74.8 | 59.7 | 88.8 | 83 | 82.4 | 82.7 | 70.5 | 97.4 |

Table 4. Quantitative comparisons on SYSU-CD and Google-CD datasets.

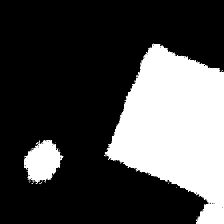
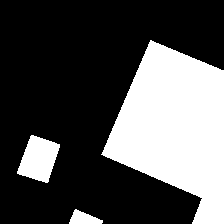
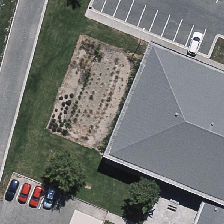
**5.2.2. Quantitative Comparisons**

Comparative results are presented in Table 3 and Table 4. The FTN method's performance is measured across various metrics on each dataset. On the LEVIR dataset, our method achieves a precision of 90.15%, recall of 82.73%, F1 score of 86.28%, IoU of 75.87%, and OA of 98.68%. Moving to the WHU dataset, our method demonstrates even higher performance with a precision of 93.51%, recall of 89.81%, F1 score of 91.62%, IoU of 84.54%, and OA of 99.33%. Similarly, on the GZ dataset, our method maintains strong performance, achieving a precision of 83.03%, recall of 82.41%, F1 score of 82.72%, IoU of 70.53%, and OA of 97.44%. Finally, on the SYSU dataset, our method achieves a precision of 79.51%, recall of 70.56%, F1 score of 74.77%, IoU of 59.70%, and OA of 88.82%. To illustrate the visual effect, displayed some typical CD results on the four datasets, as shown below.

Image A Image B Label Predicted

****

LEVIR-CD DATSET

Image A Image B Label Predicted 

WHU-CD DATSET

Figure 6: Images illustrating the actual change

and the predicted change by the pre-trained models.

**5.3. DDPM-CD: Denoising Diffusion Probabilistic Models as Feature**

**Extractors for Change Detection**

The study compares the DDPM-CD method with several state-of-the-art methods for remote sensing change detection. Notably, it includes Fully-Convolutional Early-Fusion (FC-EF), Fully-Convolutional Siamese-Difference (FC-Saim-diff), Fully-Convolutional Siamese-Concatenation (FC-Siam-conc), Dual-Task Constrained Siamese Network (DT-SCN), Spatial-Temporal Attention Network (STA-Net), Densely Connected Siamese Network (SNU-Net), Bi-Temporal Image Transformer (BIT), and Transformer-based Siamese Network (Change-Former). BIT and Change-Former employ transformers, considered the latest SOTA methods. FC-EF, FC-SD, and FC-SC start training from random initialization, while DT-SCN uses a pre-trained ResNet with Squeeze-and-Excitation Networks (SE-ResNet) from ImageNet-1k. STANet and IFNet also utilize pre-trained ResNet-18 and VGG16 on ImageNet, respectively. In contrast, SNUNet starts training from random initialization without pre-trained weights. BIT uses a pre-trained ResNet-50 for feature extraction, while Change-Former initializes hierarchical transformer blocks with pre-trained weights on ADE20k before fine-tuning for change detection.

**5.3.1. Quantitative change detection results**

Table 5 compares the change detection performance of the proposed DDPM-CD method with state-of-the-art methods, using F1, IoU, and OA scores. DDPM-CD shows significant improvement over existing methods, especially those starting from random initialization or using supervised or self-supervised pre-trained backbones. While many methods utilize pre-trained weights from ImageNet, which may not be ideal for remote sensing, DDPM-CD outperforms recent approaches relying on self-supervised pre-training. Importantly, DDPM-CD solely uses off-the-shelf remote sensing images for pre-training, eliminating the need for large classification datasets. This distinguishes it from methods like BIT and Change-Former, which rely on supervised pre-training with datasets like ImageNet and ADE20k. The substantial performance improvement validates the effectiveness of pre-trained diffusion models for generating meaningful representations in change detection.

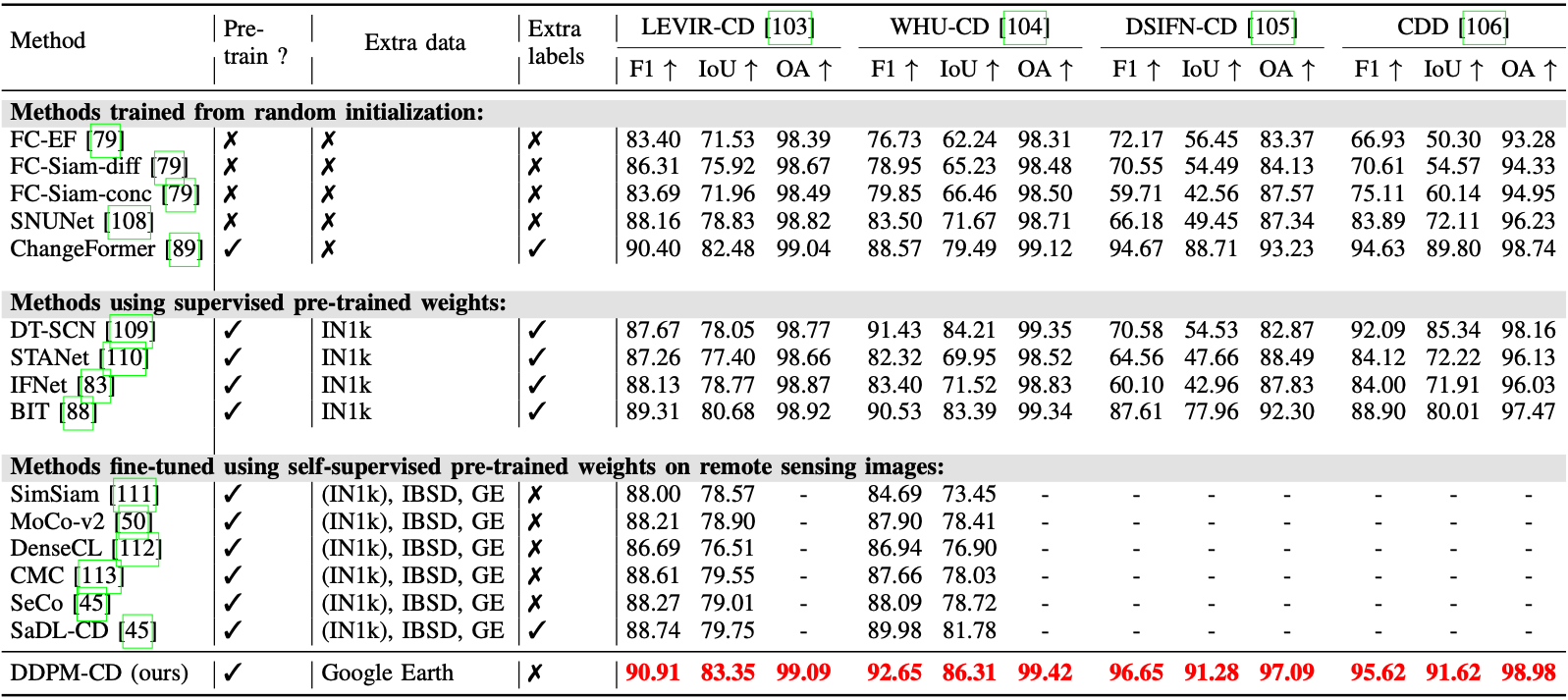


TABLE 5: The average quantitative change detection results on the

LEVIR-CD, WHU-CD, DSIFN-CD, and CDD **test** sets. Authors DDPM-CD achieves the best results in F1, IoU, and OA metrics (highlighted in red colour) on all four datasets. ↑ indicates higher the value better the change detection performance. “-” indicates not reported or not available to us. (IN1k) indicates pre-training process is initialized with the ImageNet pre-trained weights. IN1k, IBSD, and GE refers to ImageNet1k , Inria Building Segmentation Dataset, and Google Earth.

**5.3.2. Qualitative change detection results**

Qualitative change detection results are provided alongside quantitative metrics, showcasing the effectiveness of DDPM-CD compared to state-of-the-art methods across various datasets. Examples from LEVIR-CD, WHU-CD, DSIFN-CD, and CDD datasets illustrate the superior performance of DDPM-CD in capturing changes accurately with fewer false positives and false negatives. In LEVIR-CD, DDPM-CD outperforms other methods in detecting building changes, even in complex scenarios. Similarly, in WHU-CD, DDPM-CD demonstrates better accuracy, especially in detecting challenging changes like building shadows and large-scale transformations. For general change detection datasets like DSIFN-CD and CDD, DDPM-CD excels in identifying intricate changes such as highways and narrow roadways, surpassing the limitations of existing methods. These qualitative comparisons highlight DDPM-CD's effectiveness in delivering robust features for accurate change detection, underscoring its superiority over state-of-the-art methods.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Time step t | LEVIR-CD |  |  |  | WHU-CD |  |  |  | DSIFN-CD | | |  | CDD | | |
|  | F1 | IoU | OA |  | F1 | IoU | OA |  | F1 | IoU | OA |  | F1 | IoU | OA |
| 5 | 89.70 | 81.40 | 99.20 |  | 91.60 | 84.50 | 99.20 |  | 93.90 | 88.40 | 96.10 |  | 91.20 | 83.90 | 91.20 |
| 50 | 90.70 | 82.90 | 99.20 |  | 92.70 | 86.50 | 99.30 |  | 94.20 | 89.00 | 96.30 |  | 93.80 | 88.30 | 98.60 |
| 100 | 90.50 | 82.70 | 99.20 |  | 92.80 | 86.50 | 99.30 |  | 95.00 | 90.40 | 96.80 |  | 94.30 | 89.30 | 98.70 |
| 150 | 90.10 | 82.00 | 99.20 |  | 92.30 | 85.80 | 99.30 |  | 94.60 | 89.70 | 96.50 |  | 94.30 | 89.30 | 98.70 |
| 50, 100 | **91.00** | **83.50** | **99.30** |  | 93.10 | 87.10 | 99.30 |  | 94.50 | 89.60 | 96.50 |  | 94.90 | 90.30 | 98.80 |
| 50, 100, 400 | 91.30 | 83.90 | 99.30 |  | 93.50 | 87.80 | 99.40 |  | 95.40 | 91.20 | 94.10 |  | 95.60 | 91.60 | 99.00 |
| 50, 100, 650 | 91.10 | 83.70 | 99.30 |  | 93.00 | 87.00 | 99.30 |  | 95.10 | 90.60 | 96.90 |  | 95.20 | 90.90 | 98.00 |

TABLE 6: Results of the authors performing

the change detection on the val-set of LEVIR-CD, WHU-CD, DSIFN-CD, and CDD.

Implementation of the model from the DDPM pre-trained model was on the initial stage and the dataset for the pretraining was being collected.

**CONCLUSION**

In summary, this internship report presents a comprehensive exploration of cutting-edge methodologies in deep learning and computer vision tailored for satellite imagery analysis. Through meticulous examination of three seminal papers, including the development of Change-aware Contrastive Loss (CaCo), the innovative Fully Transformer Network (FTN), and the integration of Denoising Diffusion Probabilistic Models (DDPMs), significant strides have been made in addressing the complexities of remote sensing applications. The demonstrated relative improvements in various downstream tasks underscore the profound impact of these advancements, promising heightened accuracy and efficiency in change detection and feature representation. This report not only contributes substantially to the ongoing discourse in remote sensing and computational analysis but also signifies a pivotal step towards leveraging satellite imagery to effectively monitor and tackle global challenges.

**FUTURE SCOPE**

* Advanced Transformer Architectures: Exploration of more efficient and tailored Transformer architectures to further enhance computational efficiency and model performance in satellite imagery analysis.
* Unsupervised and Weakly-Supervised Learning: Development of unsupervised or weakly-supervised learning methods to alleviate the burden of manual labeling in remote sensing image datasets, thereby facilitating scalability and generalization.
* Semantic Understanding: Integration of semantic understanding techniques to extract higher-level contextual information from satellite imagery, enabling deeper insights into global phenomena such as land-use changes and environmental shifts.
* Real-Time Change Detection: Advancement towards real-time change detection systems leveraging streaming satellite imagery and rapid processing algorithms, facilitating timely responses to dynamic environmental events.
* Cross-Domain Adaptation: Exploration of cross-domain adaptation methods to enhance model generalization across diverse geographical regions and environmental conditions, ensuring robust performance in varied settings.
* Ethical Considerations: Critical examination of the ethical implications of satellite imagery analysis, including issues of privacy, surveillance, and equitable access to technology, to ensure responsible and equitable deployment of these technologies.

**REFERENCES**

1. R. Caye Daudt, B. Le Saux, A. Boulch, and Y. Gousseau. Urban change detection for multispectral earth observation using convolutional neural networks. In IGARSS, 2018.
2. Aysim Toker, Lukas Kondmann, Mark Weber, Marvin Eisenberger, Andr´es Camero, Jingliang Hu, Ariadna Pregel Hoderlein, C， a˘glar S， enaras, Timothy Davis, Daniel Cremers, et al. Dynamicearthnet: Daily multi-spectral satellite dataset for semantic change segmentation. In CVPR, 2022.
3. Oscar Manas, Alexandre Lacoste, Xavier Gir´o-i Nieto, David Vazquez, and Pau Rodriguez. Seasonal contrast: Unsupervised pre-training from uncurated remote sensing data. In ICCV, 2021.
4. Utkarsh Mall, Bharath Hariharan and Kavita Bala. "Change-Aware Sampling and Contrastive Learning for Satellite Images". In CVPR, 2023.
5. Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. CoRR,2018.
6. Fully Transformer Network for Change Detection of Remote Sensing Images  
   ***Tianyu Yan, Zifu Wan, Pingping Zhang***; Proceedings of the Asian Conference on Computer Vision (ACCV), 2022, pp. 1691-1708.
7. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: IEEE/CVF International Conference on Computer Vision. pp. 10012\_10022 (2021).
8. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 2117\_2125 (2017).
9. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 39(12), 2481\_2495 (2017).
10. M. Drusch, U. Del Bello, S. Carlier, O. Colin, V. Fernandez, F. Gascon, B. Hoersch, C. Isola, P. Laberinti, P. Martimort et al., “Sentinel-2: Esa’s optical high-resolution mission for gmes operational services,” Remote sensing of Environment, vol. 120, pp. 25–36, 2012.
11. N. Gorelick, M. Hancher, M. Dixon, S. Ilyushchenko, D. Thau, and R. Moore, “Google earth engine: Planetary-scale geospatial analysis for everyone,” Remote sensing of Environment, vol. 202, pp. 18–27,2017.
12. C. Saharia, J. Ho, W. Chan, T. Salimans, D. J. Fleet, and M. Norouzi “Image super-resolution via iterative refinement,” arXiv preprint arXiv:2104.07636, 2021.
13. Chen, H., Shi, Z.: A spatial-temporal attention-based method and a new dataset for remote sensing image change detection. Remote Sensing 12(10), 1662 (2020).
14. Ji, S., Wei, S., Lu, M.: Fully convolutional networks for multisource building extraction from an open aerial and satellite imagery data set. IEEE Transactions on Geoscience and Remote Sensing 57(1), 574\_586 (2018).
15. Shi, Q., Liu, M., Li, S., Liu, X., Wang, F., Zhang, L.: A deeply supervised attention metric-based network and an open aerial image dataset for remote sensing change detection. IEEE Transactions on Geoscience and Remote Sensing 60, 1\_16 (2021)
16. Liu, M., Shi, Q., Marinoni, A., He, D., Liu, X., Zhang, L.: Super-resolution-based change detection network with stacked attention module for images with di\_erent resolutions. IEEE Transactions on Geoscience and Remote Sensing 60, 1\_18(2021)
17. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A largescale hierarchical image database. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 248\_255 (2009)
18. C. Zhang, P. Yue, D. Tapete, L. Jiang, B. Shangguan, L. Huang, and G. Liu, “A deeply supervised image fusion network for change detection in high resolution bi-temporal remote sensing images,” ISPRSJournal of Photogrammetry and Remote Sensing, vol. 166, pp. 183–200, 2020.
19. M. Lebedev, Y. V. Vizilter, O. Vygolov, V. Knyaz, and A. Y. Rubis,“Change detection in remote sensing images using conditional adversarial networks.” International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences, vol. 42, no. 2, 2018.
20. Wele Gedara Chaminda Bandara, Nithin Gopalakrishnan Nair, and Vishal M Patel. Ddpm-cd: Remote sensing change detection using denoising diffusion probabilistic models. CoRR, 2022.