UNIVERSITY OF TARTU
Faculty of Science and Technology
Institute of Computer Science
Computer Science Curriculum

Prakhar Srivastava

Optics-free Image Classification with Deep Metric Learning

Master's Thesis (30 ECTS)

Supervisor: Kallol Roy, PhD

Tartu 2023

Optics-free Image Classification with Deep Metric Learning

Abstract:

The lens is defined as a device to control how light reaches the imaging surface and is a fundamental component of imaging. Imaging applications are ubiquitous, ranging from autonomous driving to biomedical applications. With advancements in imaging technology, new applications in fields such as biomedicine and defense are driving a significant push toward the miniaturization of cameras. Unfortunately, this miniaturization has a fundamental difficulty: the total amount of light collected at the sensor image decreases with the lens aperture. As a result, ultra-miniature images collected simply by scaling down the optics and sensors are noisy. Thus, an innovative concept is introduced in which the lens is removed; however, the resulting images obtained without the lens are degraded. To address this issue, an alternative encoding scheme and computational algorithms are used to retrieve back the image, which we refer to as optics-free imaging.

This thesis proposes a novel approach to using Deep Metric Learning for optics-free image classification. Our Deep Convolutional Neural Network uses the image similarity metric for its learning algorithms. In this thesis, first, we trained our model with the dataset composed of degraded Cifar10 images taken in the lab and the original Cifar10 dataset for image classification. In the next task, we train our model on optics-free (reconstructive) Cifar10 images obtained from degraded Cifar10 images using CycleGAN, along with original Cifar10 images for image classification. Finally, we employed Bayesian Prior to update the learning and computed the KL divergence.

Keywords: Optics-free image classification, Deep Metric Learning, Triplet loss, Quadruplet loss, Bayesian inference

CERCS: P176-Artificial intelligence; T111-Imaging, image processing

Optikavaba kujutiste klassifikatsioon sügava meetrilise õppimisega

Lühikokkuvõte:
Objektiiv on määratletud kui seade, mis kontrollib valguse jõudmist kujutise pinnale ja on pildistamise põhikomponent. Pildindusrakendusi on kõikjal, alates autonoomsest sõidust kuni biomeditsiiniliste rakendusteni. Tänu pilditehnoloogia edusammudele ajendavad uued rakendused sellistes valdkondades nagu biomeditsiin ja kaitse olulisel määral kaamerate miniaturiseerimise suunas. Kahjuks on sellel miniaturiseerimisel põhiline raskus: sensori kujutisele kogutud valguse koguhulk väheneb koos objektiivi avaga. Selle tulemusena on üliminiatuursed pildid, mis on kogutud lihtsalt optika ja andurite vähendamisega, mürarikkad. Seega võetakse kasutusele uuenduslik kontseptsioon, mille puhul objektiiv eemaldatakse; ilma objektiivita saadud kujutised on aga halvenenud. Selle probleemi lahendamiseks kasutatakse kujutise taastamiseks alternatiivset kodeerimisskeemi ja arvutusalgoritme, mida me nimetame optikavabaks pildistamiseks.

See lõputöö pakub välja uudse lähenemisviisi Deep Metric Learning kasutamiseks optikavaba kujutiste klassifitseerimiseks. Meie sügav konvolutsiooniline närvivõrk kasutab oma õppimisalgoritmide jaoks pildi sarnasuse mõõdikut. Selles lõputöös koolitasime esiteks oma mudelit andmestikuga, mis koosnes laboris tehtud halvenenud Cifar10 piltidest ja originaalsest Cifar10 andmekogumist piltide klassifitseerimiseks. Järgmises ülesandes treenime oma mudelit optikavabade (rekonstrueerivate) Cifar10 piltide jaoks, mis on saadud halvenenud Cifar10 piltidest, kasutades CycleGAN-i, koos originaalsete Cifar10 piltidega piltide klassifitseerimiseks. Lõpuks kasutasime õppimise värskendamiseks Bayesian Priorit ja arvutasime KL-i erinevuse.

Keywords: Optikavaba kujutise klassifikatsioon, sügav meetriline õpe, kolmiku kadu, neljandiku kadu, bayesi järeldus

CERCS: P176-Tehisintellekt; T111-Kujutised, pilditöötlus

Contents

Acknowledgements

I am grateful for the help and support I received from various individuals and organizations during my thesis project. Firstly, I want to thank my supervisor, Kallol Roy, for his constant guidance and motivation, which greatly influenced the success of my thesis research and helped me to achieve my goals. In addition, I am grateful to Prof. Rajesh Menon and his team at the University of Utah for providing me with essential data and research material, which significantly contributed to the quality and depth of my research. I'd also like to thank my parents, brother, and sister-in-law for their unwavering support and understanding throughout my academic career. Their love and support have given me strength and motivation. Finally, I'd like to thank the University of Tartu for providing me with the resources and facilities I needed to conduct my research. I am grateful for the valuable knowledge and skills I gained while attending university. I am honored and fortunate to have worked with such incredible individuals and organizations. Thank you for your contributions and encouragement.

1 Introduction

1.1 Miniaturized Camera Models

In both cameras and the human eye, a lens is a transparent object with curved surfaces that refracts light, bending it to converge or diverge at a certain point. In the eye, the lens is located behind the pupil and is crucial in focusing light onto the retina, where the image is detected and transmitted to the brain. Similarly, conventional cameras rely on a series of lenses and other optical elements to capture and focus light onto a digital sensor or film. The major drawback with these conventional cameras is the size which is why these cameras cannot be used in the fields like biomedical, surveillance, industrial inspection and microscopy, etc. This has led to the development of miniaturized cameras to provide a convenient and versatile imaging solution for a wide range of applications like surveillance, medical imaging, automotive cameras, drones, wearable devices, smartphones, etc as shown in Figure 1.

Figure 1. Miniature Camera usage

However, all of the necessary optical elements, including lenses, are difficult to fit into a tiny package. Due to this, maintaining image resolution along with reducing the size of a camera's components can result in a decrease in image quality. Additionally, miniaturized cameras also struggle with issues such as noise, distortion, and limited dynamic range. The potential areas where lensless imaging will find applications:

1. Biomedical imaging: Lensless imaging can revolutionize biomedical imaging by allowing for high-resolution imaging of cells and tissues without using bulky and expensive lenses. This develops into new cellular processes and disease states, as well as improved diagnostic and treatment options.

2. Microscopy: Lensless imaging will be used in microscopy to capture high-resolution images of small objects, such as bacteria or nanoparticles. This could be useful in fields such as materials science, biology, and nanotechnology.

3. Consumer electronics: Lensless imaging could be used in various consumer electronic devices, such as smartphones, tablets, and wearable technology. By eliminating the need for lenses, lensless imaging could allow for smaller, lighter, and more affordable devices.

4. Security and surveillance: Lensless imaging could be used in security and surveillance applications, such as facial recognition and biometric scanning. It could also be used to capture images in low-light conditions or in environments where traditional cameras may not be practical.

5. Industrial inspection: Lensless imaging could be used in industrial inspection applications, such as inspecting the surface of manufactured components or detecting defects in materials.

1.1.1 Optics-Free Imaging

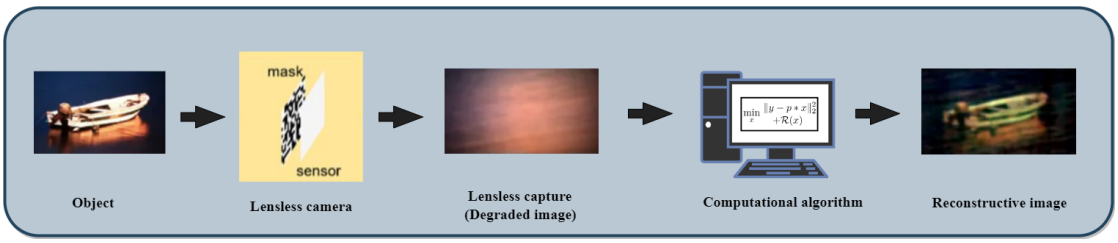


Figure 2. Optics-Free/ Lensless Imaging [BARV20]

Optics-free(Lensless) imaging eliminates the need for lenses and other optical elements. A computational algorithm is used for encoding is used instead. Instead, it relies on capturing the diffraction pattern created by an object when light is shone on it. The digital senso captures the diffraction pattern. This diffraction pattern is then processed using computational algorithms[BARV20] to reconstruct an image, as shown in Figure 2. The several advantages of using Optics-free or Lensless imaging that drives research and innovation:

1. Lensless cameras are less expensive and light weighted.

2. By eliminating mechanical components like lenses, lensless cameras can be more reliable and have a longer lifespan.

3. Lensless imaging is highly customizable and adaptable to different imaging requirements.

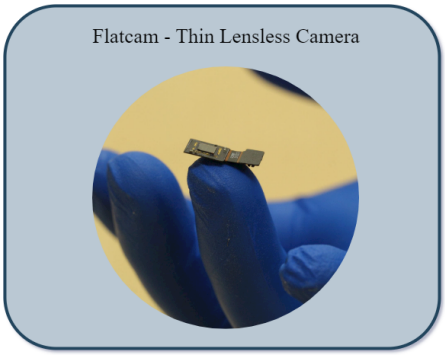


Figure 3. Lensless Camera

Though there are many advantages to optics-free imaging, they do have some technical bottlenecks of high computational requirements and environmental sensitivity. Lensless imaging systems rely on computational algorithms to reconstruct images from the diffraction patterns captured by the digital sensor. These algorithms are often complex and require significant computing power, making lensless imaging systems more expensive and challenging to implement than traditional lens-based systems. The computational requirements also mean that the processing time required to generate an image can be longer, which may not be suitable for applications that require real-time image processing. Lensless imaging systems can be sensitive to changes in the environment, such as changes in lighting conditions or the presence of dust or other particles in the air. This is because the diffraction pattern captured by the digital sensor is affected by any objects in its path, including dust particles. Any changes in lighting conditions can also affect the diffraction pattern, leading to inaccuracies in the reconstructed image. Additionally, the lack of lenses in lensless imaging systems means they are more susceptible to aberrations caused by environmental factors such as temperature changes or vibrations, which can further impact image quality. This motivates us to use a machine learning approach on the images captured through the optics-free method.

1.2 Feature Learning on Optics-Free Images

The spectacular success of deep learning comes from automatic feature learning by deep neural nets during end-to-end training. Recently different research groups have been using deep learning architectures to learn the important features embedded in noise[ZLT$^+$22]. The noise in the images comes from various sources, e.g., encoding, storage, etc. Recently CycleGAN has been used for denoising the optics-free images as shown in Fig 6. In CycleGAN, the models learn the essential features from a game-theoretic perspective[MMBJ$^+$21]. The additional constraint for the learning algorithms in optics-free imaging is the scarcity of training samples and lower-resolution images.

Bayesian Priors and hybrid approaches are the way to tackle these limitations.

This thesis proposes a novel approach to deep metric learning for classifying lensless images. The proposed model uses an image similarity metric that can be calculated in the Euclidean or Riemannian manifold[AL20], making it invariant to various noise characteristics. During training, the model is fed with pairs of images: a noisy optics-free image and its corresponding cleaner version. The invariant features between the image pairs are learned through gradient descent during its training. We use the Lensless and Reconstructive Cifar10 dataset for training. This dataset comprises ten different classes and a vast number of labeled images, making it a suitable selection for training and testing our image classification models.

2 Literature Survey

Ganghun Kim et. al [KKPM17] investigates image classification using a MINIST lensless dataset. The dataset is prepared by a bare CMOS sensor that captures grayscale images of handwritten digits (0-9) of the MINIST dataset[Den12] from an LCD screen. Different models of SVM, decision tree, and k-nearest neighbors are used for the classification. Their approach achieves a high accuracy of 99 percent for classifying only 0 and 1 digits from lensless images, but as they expand to other digits, the accuracy gradually decreases. Jasper Tan et. al [TNA$^+$19] solve face detection and verification using lensless cameras. The authors have used Faster R-CNN for face detection and a convolutional neural network (CNN) for face verification. The models are trained from reconstructed images instead of raw lensless images. Eric Bezzam et.al [BVS22] group investigates on the cost-effective lensless imaging system for reliable and privacy-preserving classification. The author uses optimum optical encoding as a regularization for generating trained embeddings. These embeddings are fed to FCNN architecture for classification. The experiments were conducted on the MINIST dataset[Den12], which is limited to digits. While the end-to-end training process has been successful, the paper also acknowledges that it can be computationally expensive due to optical wave propagation simulations. Additionally, relying on physical devices for computation has some drawbacks, including susceptibility to degradation and device tolerances, particularly for low-cost components. Other papers like [KSB$^+$20] ,[MYK$^+$19]and[RKJ21] all discuss image reconstruction techniques using either deep Learning or traditional image retrieval approaches. [KSB$^+$20] employs a neural network-based method to learn natural scene statistics and produce photorealistic scene reconstructions from lensless measurements. In contrast,[MYK$^+$19] presents a reconstruction algorithm that uses a sequence of measurements acquired through a sparse set of masks to reconstruct the scene. [RKJ21] proposes an approach that accurately reconstructs the scene by estimating the point spread function (PSF) from the measured data. Despite their differences, these papers share certain limitations. Firstly, all three methods require a significant number

of measurements, which can be time-consuming and computationally intensive. This can pose a challenge in real-world applications where time and resources are limited. Additionally, each approach has certain constraints or assumptions that may not hold true in practical scenarios. For instance, [KSB$^+$20] does not explicitly model the PSF, which can limit its performance under low-light conditions or when the object is close to the sensor. The reconstruction algorithm in [MYK$^+$19] assumes that the masks used for measurement are well-designed and do not introduce significant noise or artifacts. Lastly, the PSF estimation in [RKJ21] assumes spatial invariance, which may not always be the case in practical scenarios.

3 Data Acquisition and Prepossessing

We conducted our experiments on Cifar10 dataset[KNH]. Two additional datasets prepared by the Computational Imaging research group of the University of Utah are also used. Our experiments are divided by the two pairs of datasets:

1. Lensless Cifar10 dataset and Original Cifar10 dataset[KNH]
2. Reconstructed Cifar10 dataset[NM22a]and Original Cifar10 dataset[KNH]

3.1 Original Cifar10 dataset

The CIFAR-10 dataset[KNH] is a collection of 60,000 32x32 color images grouped into 10 classes, with 6,000 images per class. The dataset is split into 50,000 training images and 10,000 testing images. The classes are: airplane, automobile , bird, cat, deer, dog frog, horse, ship , truck. Since it offers a difficult benchmark for image classification tasks, the CIFAR-10 [KNH] is frequently utilized in machine learning and computer vision research. It is a challenging dataset to accurately identify due to the small size of the images and the high number of classes. The 2009-created dataset has since grown to rank among the most used datasets for evaluating image classification techniques. Images were gathered from many sources, including the ImageNet dataset, and were accurately labeled by humans. The relatively poor resolution and noise of the images provide one of the limitations of CIFAR-10, making it challenging for models to capture fine-grained information. In this thesis, we mention this dataset as an original Cifar10 dataset [KNH]

3.2 Lensless Cifar10 dataset

Researchers used a camera (Mini-2MP-Plus, Arducam) without a lens to take pictures of images from the Cifar-10 dataset. The camera was placed either 1mm or 10mm away from a physical display of the image dataset, as shown in [Fig 4]. The display had a size

of 20 x 20 pixels when viewed from up close (1mm away), and a size of 200 x 200 pixels when viewed from a distance of 10mm. The physical size of the display was 6.22mm for the 20 x 20 pixel image and 62.2mm for the 200 x 200 pixel image. The display used was a liquid-crystal display (LCD) model named Acer G276HL, which had a resolution of 1920x1080 pixels.

Figure 4. Lensless imaging setup[NM22a]

The camera captured images without a lens by directly sensing the light coming from the display. The field of view captured by the camera was 144 degrees. The RGB image obtained from the lensless camera is of size 320×240 as shown in [Fig 5]

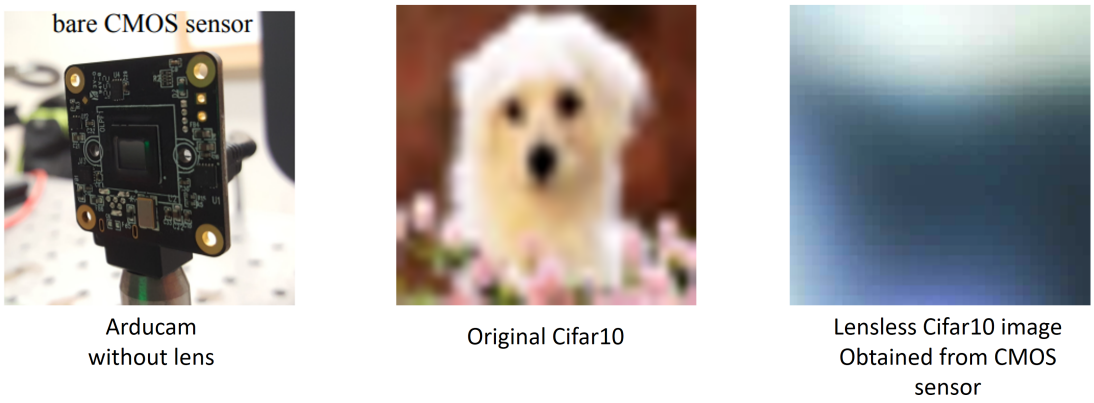


Figure 5. Right most is the lensless image captured by the Arducam using original Cifar10 image on LCD display as shown in [Fig 4]

3.3 Reconstructed Cifar10 dataset

To prepare the reconstructed CIFAR-10 dataset[NM22a], researchers have used an "optic-free" deep learning method called CycleGAN[NM22b]. The method transformed the lensless images into images that resemble the original CIFAR-10 images. It involves training two deep neural networks, a generator, and a discriminator, to learn the mapping between the two domains of images. The generator is in charge of creating reconstructed images from lensless images, whereas the discriminator distinguishes between the original and reconstructed images. To ensure high-quality images, the generator is trained to generate images that can be reverse-mapped back into their original form. This prevents any data loss during translation and is referred to as cycle consistency, as shown in [Fig 6]. However, the obtained reconstructed images are less degraded than lensless images but show a minute resemblance to the original Cifar10 images.

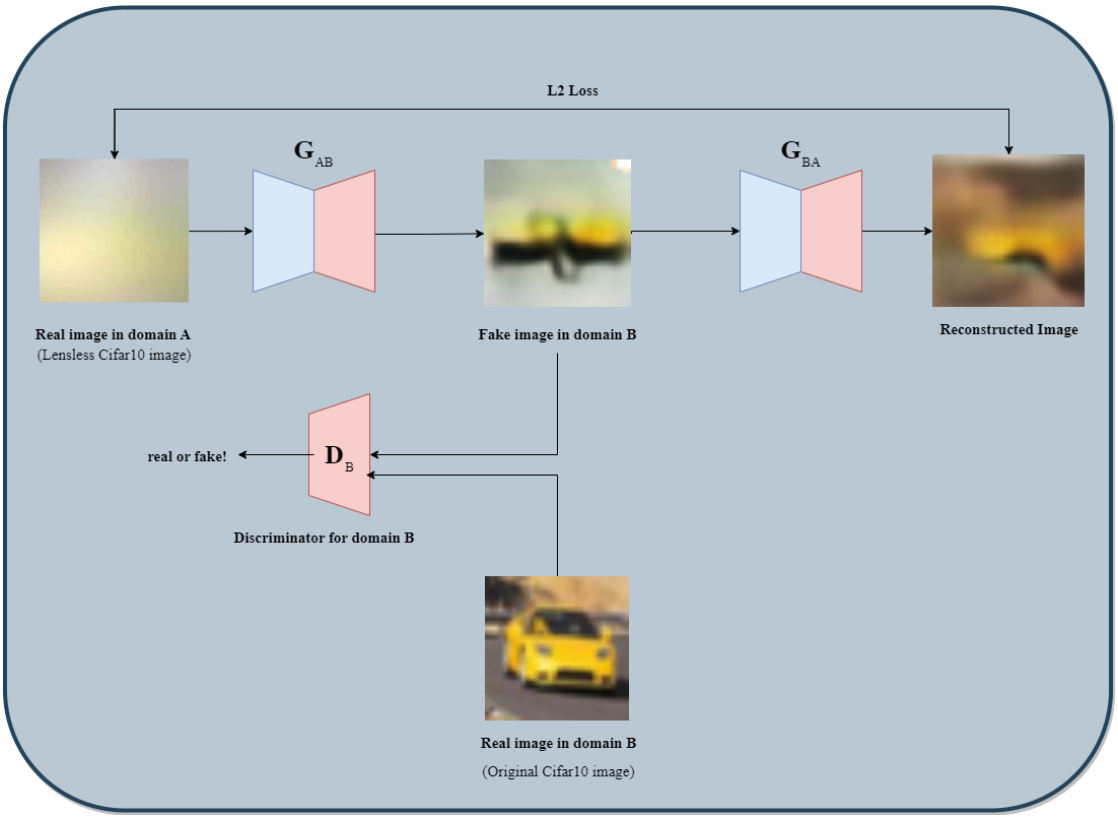


Figure 6. Reconstructed Cifar10 image using CycleGAN

3.4 Data Preprocessing

We prepare a pandas data frame(CSV file(s)) that stores the names of the files from the dataset pair that is (Lensless Cifar10 and original Cifar10 dataset) and (Reconstructed Cifar10 and original Cifar10 dataset) to train to our CNN model. This data frame has six columns which are as follows: (i) Anchor (ii) Positive (iii) Close Positive (iv) Negative (v) Label encode (vi) Label name

	anchor_img	positive_img	close_positive_img	negative_img	label	label_name
15394	cat_s_001459.jpeg	cat_s_001459.png	cat_s_001462.png	automobile_s_001307.png	3	cat
22494	fawn_s_000837.jpeg	fawn_s_000837.png	fawn_s_000844.png	pipit_s_001424.png	4	deer
10317	alauda_arvensis_s_001127.jpeg	alauda_arvensis_s_001127.png	alauda_arvensis_s_001145.png	red_deer_s_000983.png	2	bird
6895	convertible_s_000276.jpeg	convertible_s_000276.png	convertible_s_000281.png	gelding_s_001741.png	1	automobile
39343	tennessee_walker_s_001125.jpeg	tennessee_walker_s_001125.png	tennessee_walker_s_001132.png	moving_van_s_001754.png	7	horse
...	...	...	...	...	...	...
16514	felis_catus_s_000393.jpeg	felis_catus_s_000393.png	felis_catus_s_000394.png	coupe_s_000543.png	3	cat
14054	sparrow_s_000027.jpeg	sparrow_s_000027.png	sparrow_s_000028.png	wagon_s_001671.png	2	bird
24457	sika_s_000474.jpeg	sika_s_000474.png	sika_s_000475.png	passenger_ship_s_000477.png	4	deer
35950	broodmare_s_001448.jpeg	broodmare_s_001448.png	broodmare_s_001450.png	automobile_s_002579.png	7	horse
2784	jumbo_jet_s_000267.jpeg	jumbo_jet_s_000267.png	jumbo_jet_s_000268.png	bullfrog_s_000293.png	0	airplane

Figure 7. Dataframe

In these six columns in [Fig 7], each row corresponds to a specific image file from the Cifar10 dataset. The arrangement of image names is systematic, such that the "Anchor" column contains only lensless images, and its subordinates in the "Positive" column have the same image name but with the original Cifar10 image. The "Close Positive" column has images of the same class as the "Positive" column but with a different image name, while the "Negative" column contains images of a different class than the "Anchor" column and its subordinates. Additionally, there is a "Label Encode" column that stores encoded numbers ranging from 0 to 9, obtained from the Cifar10 file name in the "Anchor" column. Finally, the "Label Name" column contains the actual names of the classes corresponding to the images in the "Anchor" column.

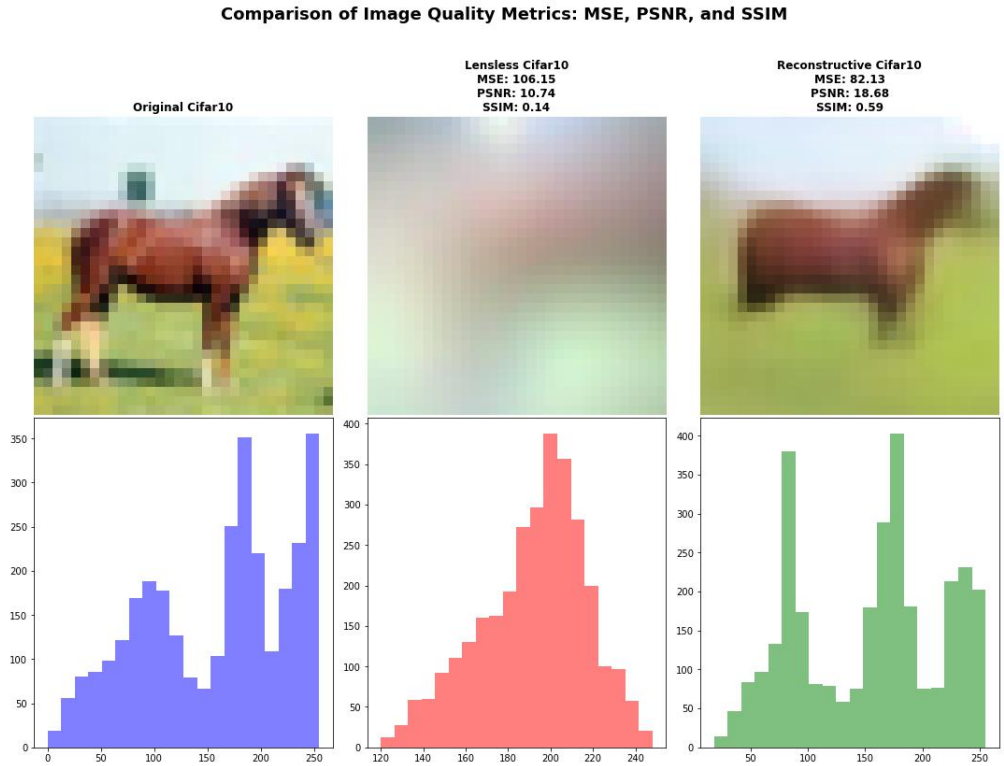


Figure 8. Comparison of Image Quality Metrics between Positive(original Cifar10) and Anchor(i.e., Lensless or Reconstructive)

We perform image similarity test between an anchor image(which is the lensless or reconstructive Cifar10 image) and the positive(which would be the original Cifar10 image) as shown in [Fig 8]. The following metrics are used:

Mean Squared Error (MSE): The MSE measures the average squared difference between the pixels in the original to the lensless image or reconstructive image. The smaller the MSE, the more similar the two images are.

Peak Signal-to-Noise Ratio (PSNR): The PSNR measures the difference between the image's noise level and the highest feasible pixel value. It is often used as a metric for image quality. The higher the PSNR, the more similar the two images are.

Structural Similarity Index (SSIM): The SSIM measures the structural similarity between the two images. It is designed to be more robust to changes in luminance and contrast. The SSIM ranges between -1 and 1, where a value of 1 indicates perfect similarity between the images, and a value of -1 indicates complete dissimilarity.

We didn't check the similarity between Anchor and Negative because it is obvious they would be different as they belong to different classes. But by checking the similarity between the anchor and the positive, we tried to understand the complexity of the problem.

After that we divided the dataframe into training, validation, and testing:

Dataframe	Number of rows
Train	39992
Validation	9998
Test	9990

We have organized our positive and negative columns based on their proximity to the anchor image class. Specifically, we have placed negative column images that are dissimilar to the anchor class far away from it. For instance, if the anchor image is of a "Deer", we would choose a "Truck" as a negative image instead of a "Horse". This is because the Horse may confuse our model due to its similarity to a four-legged animal, whereas a Truck has a distinct metallic body with tires, making it less similar to a Deer. This approach of organizing the data frame is applied to other classes as well, like cat-dog, truck-automobile, bird-airplane, and dog-horse. Additionally, we randomly split the rows from the main dataframe into three sets for training, validation, and testing. We ensure that the classes are almost equally represented in each set, which helps make our model unbiased.

4 Proposed Method

We built a convolutional neural network (CNN) and trained it using triplet and quadruplet loss functions to solve the classification problem of the lensless/reconstructive Cifar10 dataset. We augment our model with a probabilistic model of Bayesian Learning[WY16]. Bayesian prior is used to update the classification parameters. Lastly, a KL divergence method is used for evaluation. Our CNN focuses on learning feature representations that can effectively capture similarities and dissimilarities between image data points. Our approach involves training a neural network to map input data points to a high-dimensional embedding space, where we can map similar points to nearby points and dissimilar points to distant points. To accomplish this, we use the Euclidean distance[Wik] as a distance metric and train the model using triplets or quadruplets of anchor, positive, and negative points. We set a margin between the distances of the anchor and positive point and the anchor and negative point during training. Our goal is to keep the distance between the anchor and the positive point as small as possible while increasing the distance between the anchor and the negative point. This helps us learn a representation that can effectively distinguish between similar and dissimilar points in the embedding space. Our CNN model architecture[ON15] has 3 convolutional layers, 2 fully connected layers, and a softmax layer as shown in [Fig 9].

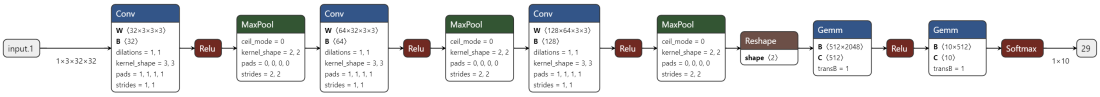


Figure 9. Convolutional Neural Network

We use a dataset that consists of two different sets of images: lensless/reconstructive Cifar10 images and original images. Specifically, we use the lensless CIFAR10 images as the anchor images, while the positive and negative images are original CIFAR10 images. The positive and negative images are from distinct classes, where the positive image shares the same class as the anchor image, while the negative image belongs to a different class.

4.1 Triplet Loss

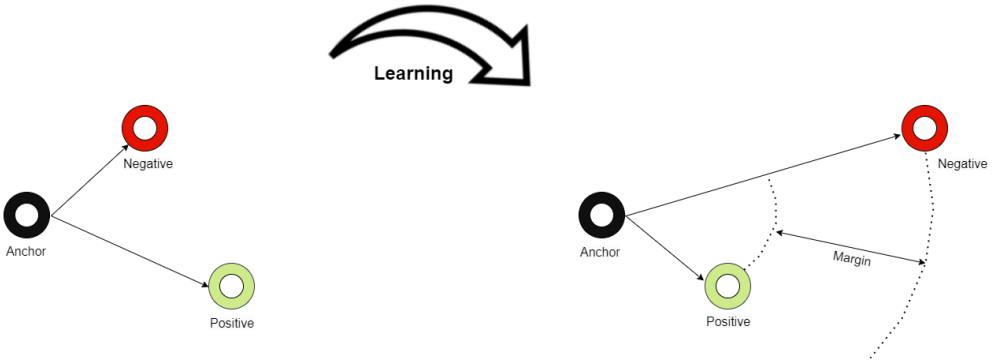


Figure 10. Triplet loss

We then use the Triplet loss function to optimize our CNN model to learn to differentiate between images of different classes in the feature embedding space. The triplet loss function works by minimizing the distance between the anchor image and the positive image while maximizing the distance between the anchor image and the negative image using the margin as shown in [Fig 10]. In addition to the Triplet loss function, we also use the Cross-entropy loss function to optimize our CNN model to learn to classify the lensless CIFAR10 images correctly into their respective classes. The cross-entropy loss function works by penalizing our model when it makes incorrect predictions and rewarding it when it makes correct predictions. By combining the Triplet loss[CGZ+16] and Cross-entropy loss[GRLGPC20] functions, we can train our CNN model to accurately classify the lensless CIFAR10 images into their respective classes.

This approach can lead to better overall performance in lensless image classification tasks. The same approach is applied for reconstructive and original Cifar10 images.

Algorithm 1: Triplet Loss Calculation

Input: Anchor tensor A, positive tensor P, negative tensor N, margin value m
Result: Triplet loss value L

EuclideanDistance(x, y) **Input:** Tensors x and y
Result: Euclidean distance between x and y
return $\sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$

triplet_loss(A, P, N, m) **Input:** Tensors A, P, and N; margin m
Result: Triplet loss L

```
// Calculate the Euclidean distances between the anchor,
   positive, and negative tensors
```

$d_{AP} \leftarrow$ euclidean_distance(A, P)
$d_{AN} \leftarrow$ euclidean_distance(A, N)

```
// Calculate the triplet loss
```

$L \leftarrow \max(d_{AP} - d_{AN} + m, 0)$

return L

Algorithm 1 shows how we train our model to recognize different objects in Lensless Cifar10 images. We take a mini-batch of images and select three from them - one we call the anchor image, another one that's from the same class as the anchor (we call this the positive image), and a third one from a different class (we call this the negative image). Our model then looks at each of these images and identifies their unique features. We calculate the triplet loss by comparing the distance between the features of the anchor and positive images with the distance between the anchor and negative images. The goal is to minimize this triplet loss by adjusting our model's parameters. This helps the model learn to tell the difference between objects in different categories and improves its overall ability to classify images accurately [HBL17].

Algorithm 2: Cross-Entropy Loss Calculation

Input: Predicted tensor y, true label tensor t
Result: Cross-entropy loss value L

softmax(x) **Input:** Tensor x
Result: Softmax output for x
return $\frac{\exp(x_i)}{\sum_j \exp(x_j)}$; `// for each` x_i

cross_entropy_loss(y, t) **Input:** Predicted tensor y, true label tensor t
Result: Cross-entropy loss value L

`// Apply the softmax function to the predicted tensor`
$p \leftarrow$ softmax(y)

`// Calculate the cross-entropy loss`
$L \leftarrow -\sum_i t_i \log(p_i)$

return L

Algorithm 2 is then used to train to classify images. The CNN has identified the unique features of each image using Algorithm 1, it feeds them into the fully connected layers. These layers produce class probabilities for each image. Next, we calculate the cross-entropy loss [GRLGPC20], which measures the difference between the true class labels and the predicted class probabilities. The goal is to minimize this cross-entropy loss by fine-tuning our model's parameters. This helps the model learn to classify lensless CIFAR-10 images accurately. By combining Algorithms 1 and 2, we train our CNN model to improve both its feature representation and classification accuracy for the lensless CIFAR-10 dataset.

4.2 Quadruplet Loss

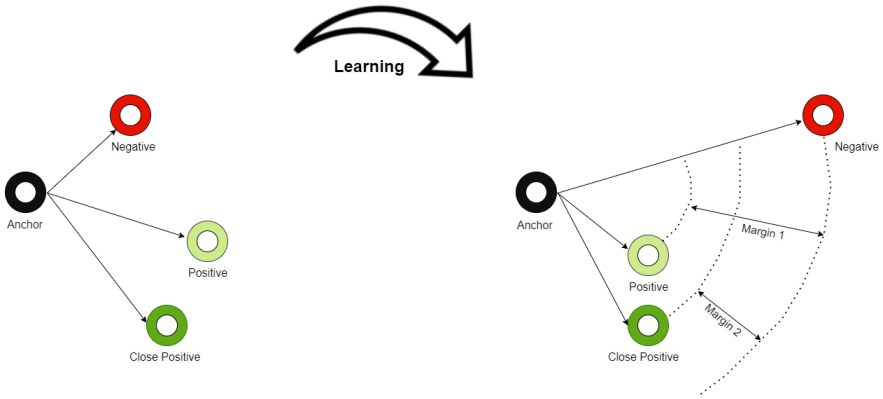


Figure 11. Quadruplet Loss

In this approach, we have combined two different loss functions, Quadruplet loss[CCZH17], and Cross-entropy[GRLGPC20], to improve the performance of our Convolutional Neural Network for classifying lensless Cifar10 images.

With the Quadruplet loss, we take four images as inputs: an anchor image, a positive image (an image from the same class as the anchor), a close positive image (an image from a different class but visually similar to the anchor), and a negative image (an image from a different class than the anchor). Here anchor would have a lensless Cifar10 image while positive, close positive, and negative would have the original Cifar10 image. The goal of this loss is to learn a feature space where the distance between the anchor and positive images is smaller than the distance between the anchor and negative images while also ensuring that the distance between the anchor and close positive images is not too small as shown in Figure 11. In addition to the Quadruplet loss, we use Cross-entropy Loss in the training process to optimize the classification task. This loss measures the dissimilarity between the predicted class probabilities and the true class probabilities, encouraging our model to correctly classify the input image into the correct class. Using the combined approach of Quadruplet loss and Cross-entropy Loss, we have found that this combination is little bit better than Triplet loss and Cross-entropy Loss. The same approach is applied for reconstructive and original Cifar10 images.

Algorithm 3: Quadruplet Loss Calculation

Input: Anchor tensor A, positive tensor P, close positive tensor C, negative tensor N, margin values m_1 and m_2
Result: Quadruplet loss value L

euclidean_distance(x, y) **Input:** Tensors x and y
Result: Euclidean distance between x and y
return $\sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$

quadruplet_loss(A, P, C, N, m_1, m_2) **Input:** Tensors A, P, C, and N; margin values m_1 and m_2
Result: Quadruplet loss L

```
// Calculate the Euclidean distances between the anchor,
   positive, close positive, and negative tensors
```

$d_{AP} \leftarrow$ euclidean_distance(A, P)
$d_{AN} \leftarrow$ euclidean_distance(A, N)
$d_{AC} \leftarrow$ euclidean_distance(A, C)

```
// Calculate the quadruplet loss
```

$L \leftarrow \max(0, m_1 + d_{AP} - d_{AN}) + \max(0, m_2 + d_{AP} - d_{AC})$

return L

In Algorithm 3, we process each mini-batch of images using a CNN model. Within each mini-batch, we select an anchor image, a positive image (of the same class as the anchor), a close positive image (of the same class as the anchor but a different image), and a negative image (of a different class from the anchor). These images are used to compute the quadruplet loss, which is a measure of the difference between the distances of the anchor and positive/close positive images and the anchor and negative image features. We train the model by minimizing this loss during the training process. After executing Algorithm 3, we employ Algorithm 2 to classify lensless CIFAR10 images, which we explained previously.

4.2.1 Quadruplet loss with causality

Algorithm 4: Quadruplet Loss with Causality Calculation

Input: Anchor tensor A, positive tensor P, close positive tensor C, negative tensor N, margin values m_1 and m_2
Result: Quadruplet loss value L

CausalQuadrupletLoss(A, P, C, N, m_1, m_2) **Input:** Tensors A, P, C, and N; margin values m_1 and m_2
Result: Causal quadruplet loss L

`// Calculate the squared Euclidean distances between the anchor, positive, close positive, and negative tensors`

$d_{AP} \leftarrow \sum_{i=1}^{n}(A_i - P_i)^2$ $d_{AN} \leftarrow \sum_{i=1}^{n}(A_i - N_i)^2$ $d_{AC} \leftarrow \sum_{i=1}^{n}(A_i - C_i)^2$

`// Calculate the causal quadruplet loss`

$L \leftarrow \max(0, m_1 + d_{AP} - d_{AN}) + \max(0, m_2 + d_{AP} - d_{AC})$

return L

In Algorithm 4 we have added a temporal component to the Quadruplet Loss by including the temporal loss term. This helps the network learn to embed images in a way that preserves the temporal causality[POW+10] between them. The temporal loss penalizes the distance between the anchor and the close positive, relative to the distance between the anchor and the positive, by at least a margin. This encourages the network to embed images in a way that preserves the temporal order of the positive and close positive images and makes them look like a sequence of images or a video. Including the temporal loss along with the similarity loss can help the network learn to distinguish between positive and negative samples while taking into account the temporal causality between them.

4.3 Bayesian Learning

Our thesis work employs Bayesian inference[Ber21], a statistical technique that updates the parameter of data generation of noisy optics-free images. We use a uniform distribution occurring in each class with 10 percent chance as a prior. We then use Bayes' theorem to combine the prior and likelihood probabilities to obtain the posterior probability occurrence of each class. The likelihood is the prediction of the CNN. Bayesian inference is used to measure the uncertainty of the each class in the data generation. By examining the posterior probabilities for each class, we can identify instances where the model is unsure about the correct classification. This information can help us enhance the model's performance and identify areas where additional data or model improvements

may be required. Overall, the implementation of Bayesian inference in our thesis work enables us to improve our understanding of the underlying distribution of each class in the CIFAR10 dataset, evaluate the quality of our model's predictions, and make more informed decisions about image classification. KL divergence[Zac21], also known as Kullback-Leibler divergence to measure the difference between two probability distributions. In our case, the two probability distribution are ground truth (of each class probability) and the updated posterior class probability distribution defined as follows:

$$KL(p||q) = \sum_i p(i) \log \left(\frac{p(i)}{q(i)} \right)$$

In our case, p represents distribution of ground truth, which is a uniform distribution over the 10 classes in the Lensless or Reconstructive CIFAR10 dataset. The posterior probability q, represent the updated probability distribution based on the observed data and the model predictions. By calculating the KL divergence between the prior distribution and the posterior distribution, we measure the amount of information gained by the model from the observed data. A lower KL divergence indicates a closer match between the prior and posterior distributions, which suggests that the model is performing well in capturing the underlying distribution of the data. A higher KL divergence indicates a larger discrepancy between the prior and posterior distributions, which suggests that the model may not be capturing the underlying distribution of the data well. Overall, the use of KL divergence in our thesis work helps us to assess the quality of our model's predictions and to identify areas where further improvements may be necessary. By monitoring the KL divergence over time, we can also track the performance of our model as it is trained on larger datasets or with more complex models.

5 Results and Discussion

This section shows the results we obtained by applying above mentioned methods. To evaluate our model performance on lensless or reconstructive images, we used evaluation metrics like Accuracy, Precision, Recall, and F1-score[B19]. These metrics provided us with a comprehensive understanding of how well our model performed in classifying lensless or reconstructive Cifar10 images. Additionally, we implemented a Bayesian inference to update our prior beliefs and calculated the KL divergence. This allowed us to gain even deeper insights into the performance of our approach.

5.1 Triplet Loss Lensless images

The hyperparameters used in the experiments are as follows: Image size = [32]; Learning rate = [0.001]; Optimizer = [Adam]; Batch size = [64]; Triplet loss Margin = [0.005];

Activation function = ['relu']; Batch normalization = False; Data augmentation = Random horizontal flip

15 Epochs:

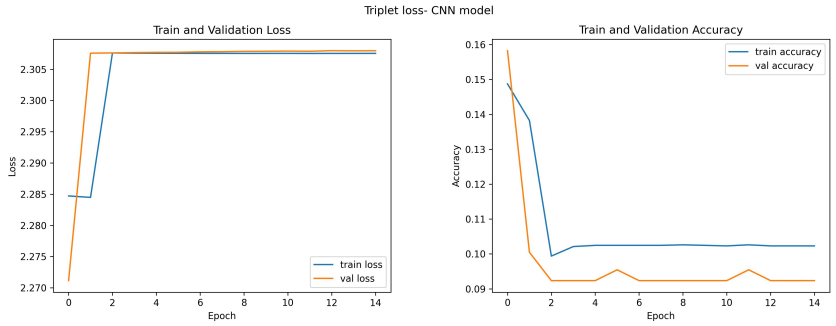


Figure 12. A CNN model trained with Triplet loss on 15 epochs

The training and validation loss graph [Fig 12]shows the model overfitted on the training data. Both the train and validation loss initially decrease but then start to increase and become constant at a value of 2.404. This behavior indicates that the model has started to memorize the training data instead of generalizing to new data.

The training and validation accuracy graph [Fig 12] shows that the model is not learning from the data, as both the train and validation accuracy initially start at a low value of 0.15 and 0.16, respectively, and then decrease before becoming constant at 0.10 and 0.09, respectively. This behavior can indicate that the model is not able to capture patterns in the data.

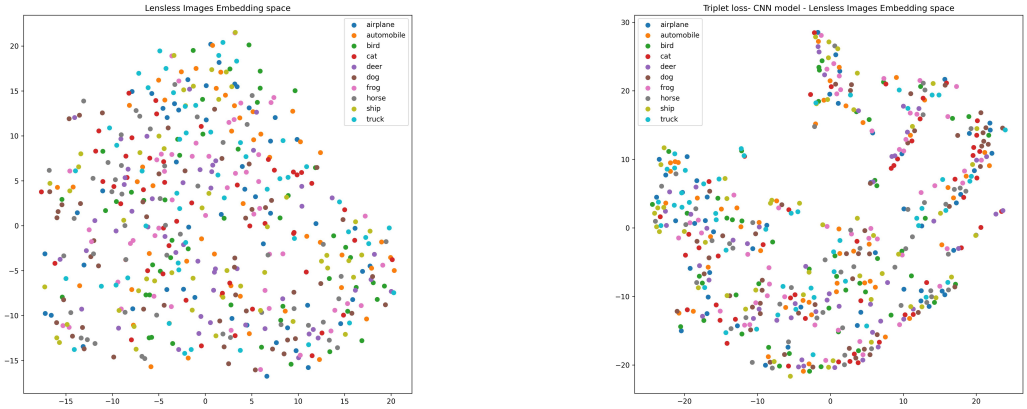


Figure 13. Lensless Images Embedding space at 15 Epochs

The plot of the test images in the embedding space shows the model struggles to make the cluster of classes [Fig 13].

30 Epochs:

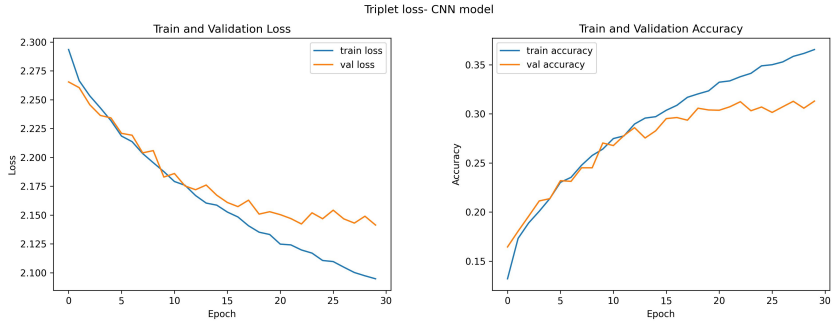


Figure 14. A CNN model trained with Triplet loss on 30 Epochs

The training and validation loss graph [Fig 14] shows that as the training progresses, the training loss decreases gradually, indicating that the model is learning from the training data and improving its predictions. The validation loss, on the other hand, starts at a lower value than the training loss, indicating that the model is overfitting to the training data. However, after 12 epochs, the train and validation loss curves start to split, with the training loss continuing to decrease while the validation loss curve starts to level off. This suggests that the model is starting to overfit the training data and is becoming less effective at generalizing to new data.

The training and validation accuracy graph [Fig 14] shows that as the training progresses, the training accuracy increases gradually, indicating that the model is getting better at predicting the correct labels for the training data. The validation accuracy, on the other hand, starts at a higher value than the training accuracy, indicating that the model is already able to generalize somewhat to new data. However, the validation accuracy increases gradually and does not reach the same level as the training accuracy. This suggests that the model is still overfitting to some extent.

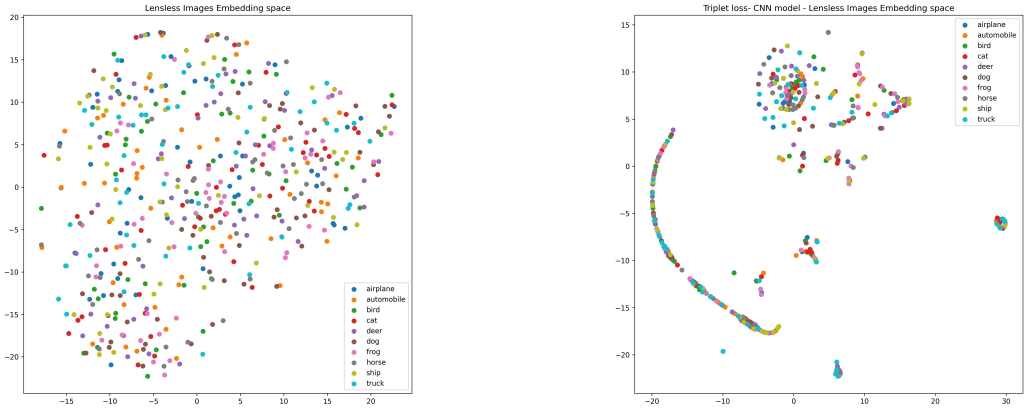


Figure 15. Lensless Images Embedding space at 30 Epochs

The plot of the test images in the embedding space shows the model is able to make clusters but struggles to make clusters of similar classes[Fig 15] .

50 Epochs:

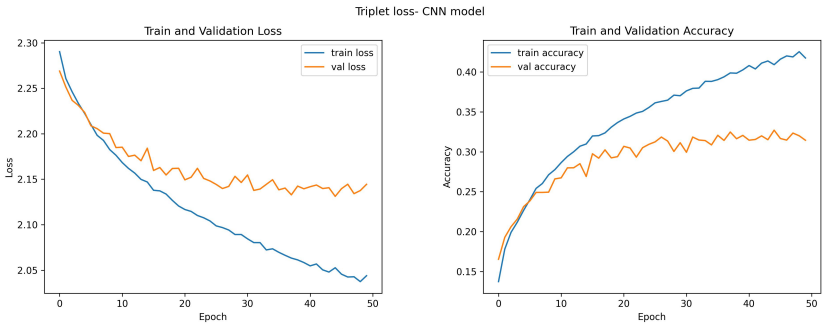


Figure 16. A CNN model trained with Triplet loss on 50 Epochs

The training and validation loss graph[Fig16] shows that after seven epochs, the gap between the train and validation loss starts to increase. This trend suggests that the model

is overfitting on the training data, i.e., the model is getting too complex and is fitting the training data too well, which results in poor generalization on the validation set.

The training and validation accuracy graph[Fig16] shows accuracies increase with each epoch, but the gap between them also increases after six epochs. This trend suggests that the model is overfitting to the training data, similar to the trend observed in the loss graph. However, unlike the loss graph, the accuracy graph shows that the model's performance is relatively good on both the training and validation sets, with validation accuracy increasing from 0.17 to 0.33 over 50 epochs.

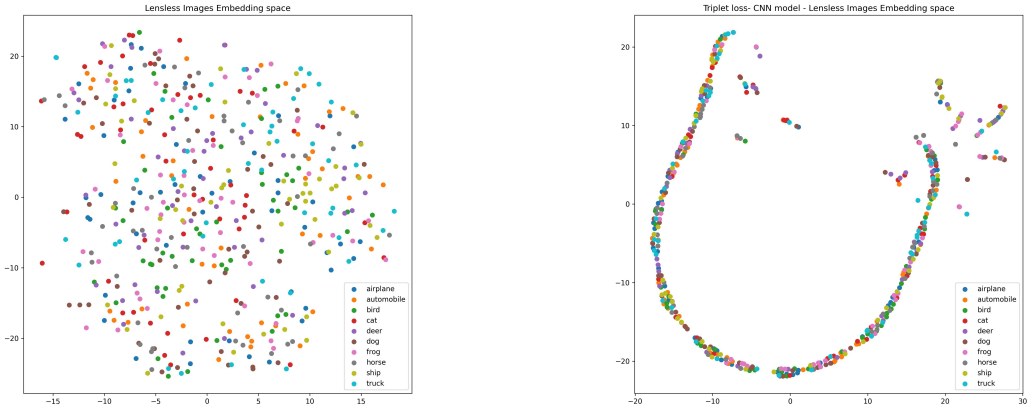


Figure 17. Lensless Images Embedding space at 50 Epochs

When we plot the test images in the embedding space, the model struggles to make similar class clusters[Fig 17].

Table 1. Triplet Loss - Lensless images

Lensless Images	Accuracy	Precision	Recall	F1score	Epochs
test images	16.40	15.47	16.40	10.92	15
test images	30.73	30.37	30.73	30.22	30
test images	31.13	31.16	31.13	30.62	50

Overall, when we give test images to our model to classify lensless Cifar10 images, the model that is trained on 50 epochs scores the highest.

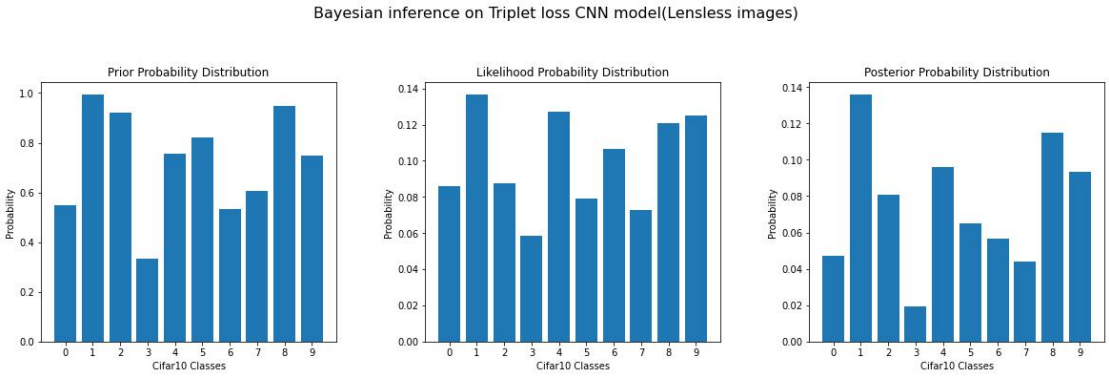


Figure 18. Bayesian inference on Triplet loss CNN model(Lensless Image)

We trained three convolutional neural networks (CNN) models with different epoch rates for image classification on the Lensless CIFAR-10 dataset. After evaluating the models, we found that model that was trained on 50 Epochs had a higher accuracy rate than the others. To further improve the performance of this model, we applied Bayesian inference and KL divergence. To begin, we calculated the prior probabilities, which represented our initial beliefs about the likelihood of each class in the CIFAR-10 dataset. We then used the trained CNN model prediction to calculate the likelihood probabilities, which represented the predicted probabilities for each class. Finally, we used Bayes' theorem to calculate the posterior probabilities, which updated our beliefs about the probability of each class based on the predicted probabilities from the model[Fig18]. We then calculated the KL divergence between the prior and posterior distributions using the posterior probabilities. The KL divergence for this model was 0.4077127916009669 nats.

This result suggests that the model's predicted probabilities were relatively close to our prior beliefs, indicating that the model was able to capture the underlying patterns in the data well. However, the fact that the KL divergence was not zero suggests that there were still some discrepancies between the model's predictions and our prior beliefs. This could indicate areas where the model could be improved, such as by incorporating additional prior knowledge or by fine-tuning the model's architecture.

5.2 Triplet Loss Reconstructive images

Hyperparameters for the following experiments are as follows: Image size = [32]; Learning rate = [0.001]; Optimizer = [Adam]; Batch size = [64]; Triplet loss Margin = [0.005]; Activation function = ['relu']; Batch normalization = True; Data augmentation = Random horizontal flip, Random vertical flip

15 Epochs:

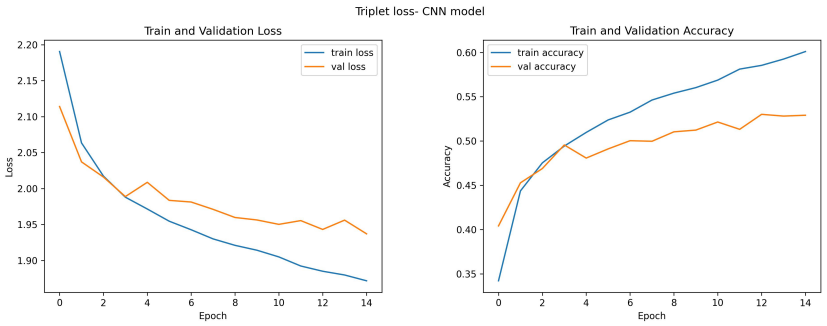


Figure 19. A CNN model trained with Triplet loss on 15 Epochs

In the train and validation loss graph[Fig19], training loss starts high and gradually decreases over the 15 epochs, indicating that the model is learning and improving over time. Similarly, the validation loss also starts high but decreases gradually over the epochs. The convergence of both the train and validation losses during the second and third epochs suggests that the model has reached a point where it can generalize well to the validation set. However, after the third epoch, the gap between the training and validation losses starts to increase, indicating that the model may be overfitting to the training data.

In the train and validation accuracy graph[Fig19], training accuracy starts low but increases steadily over the epochs, indicating that the model is learning and improving its predictions. Similarly, the validation accuracy also increases but at a slower rate than the training accuracy. The convergence of both the training and validation accuracies during the second and third epochs suggests that the model is performing well on both the training and validation data. However, after the third epoch, the gap between the training and validation accuracies starts to increase, indicating that the model may be overfitting to the training data.

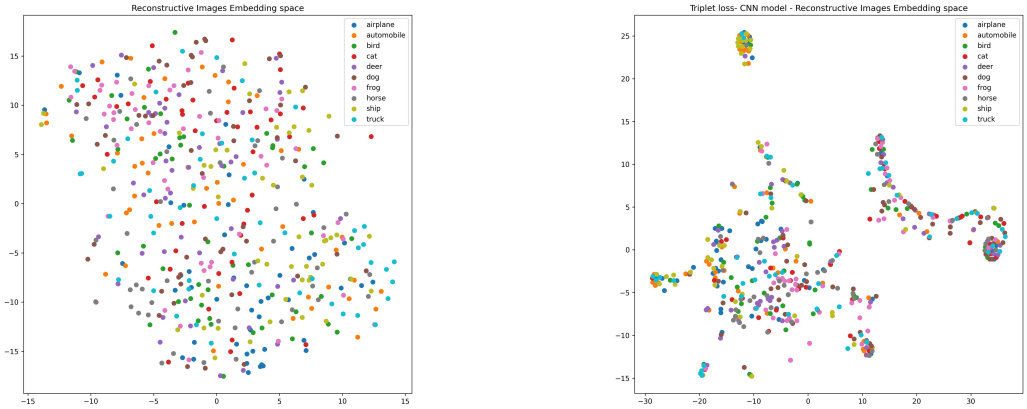


Figure 20. Reconstructive Images Embedding space at 15 Epochs

When we plotted the test images in the embedding space, the model tried to make clusters but struggled in making similar class clusters [Fig 20].

30 Epochs:

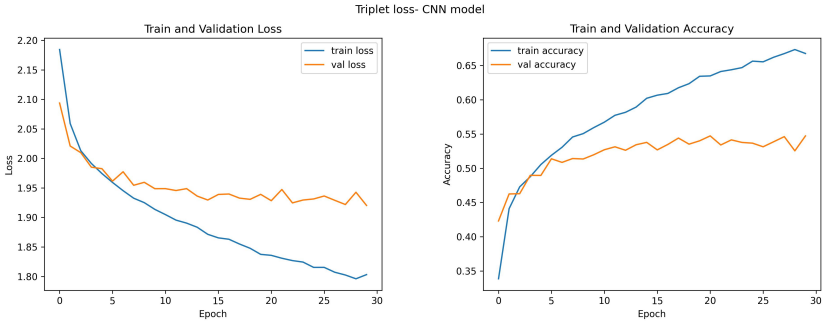


Figure 21. A CNN model trained with Triplet loss on 30 Epochs

In the train and validation loss graph[Fig21], the training loss decreases more significantly as compared to the validation loss, indicating that the model is overfitting to the training data. However, after the 5th epoch, the gap between the train and validation loss also starts to increase, indicating that the model's performance on the validation data is deteriorating. This behavior could be due to the model's inability to generalize well to unseen data.

In the Train and Validation Accuracy graph[Fig21], the training accuracy gradually increases as the number of epochs increases, while the validation accuracy increases slowly but not as much as the training accuracy. This behavior indicates that the model is

learning from the training data but not generalizing well to the validation data. As the number of epochs increases, the gap between the training and validation accuracy also increases, indicating overfitting.

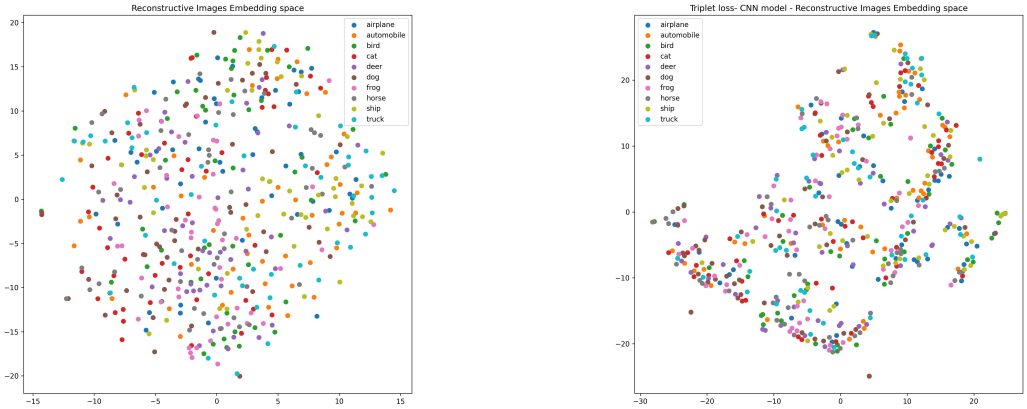


Figure 22. Reconstructive Images Embedding space at 30 Epochs

When we plot the test images in the embedding space, the model struggles to make clusters of classes[Fig 22].

50 Epochs:

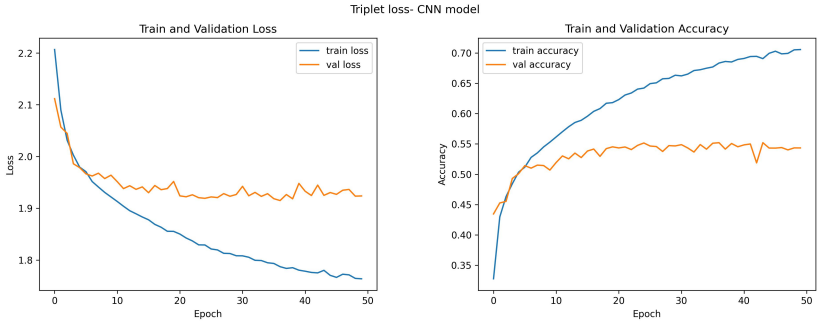


Figure 23. A CNN model trained with Triplet loss on 50 epochs

In the train and validation loss graph[Fig23], both train and validation loss decrease steadily over the 50 epochs. However, the validation loss is consistently higher than the training loss, indicating that the model is overfitting to the training data. We also see that the gap between the two losses gradually increases after epoch 6, suggesting that the model is becoming less effective at generalizing to new data as the training progresses.

In the Train and Validation Accuracy graph[Fig23], the training accuracy improves significantly from 0.33 to 0.70 over the 50 epochs. However, the validation accuracy only improves from 0.44 to 0.53 over the same period, indicating that the model is not performing well on new data. We also see that the gap between the two accuracies increases after epoch 6, further supporting the notion that the model is overfitting to the training data.

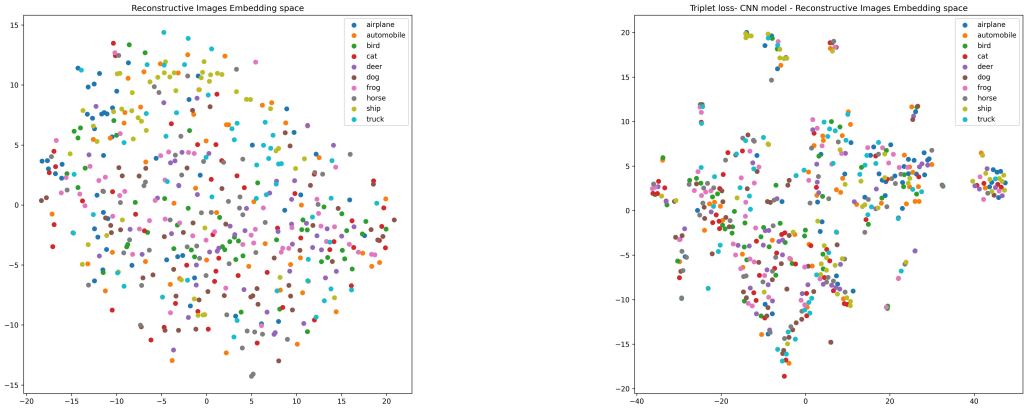


Figure 24. Reconstructive Images Embedding space at 50 Epochs

When we plot the test images in the embedding space, the model struggles to make clusters of classes [Fig24].

Table 2. Triplet Loss

Reconstructive Images	Accuracy	Precision	Recall	F1score	Epochs
test images	38.37	38.75	38.37	37.72	15
test images	39.58	39.89	39.58	39.41	30
test images	39.56	40.07	39.56	39.41	50

Overall, when we give test images to our model to classify reconstructive Cifar10 images, the model that is trained on 30 epochs scores the highest.

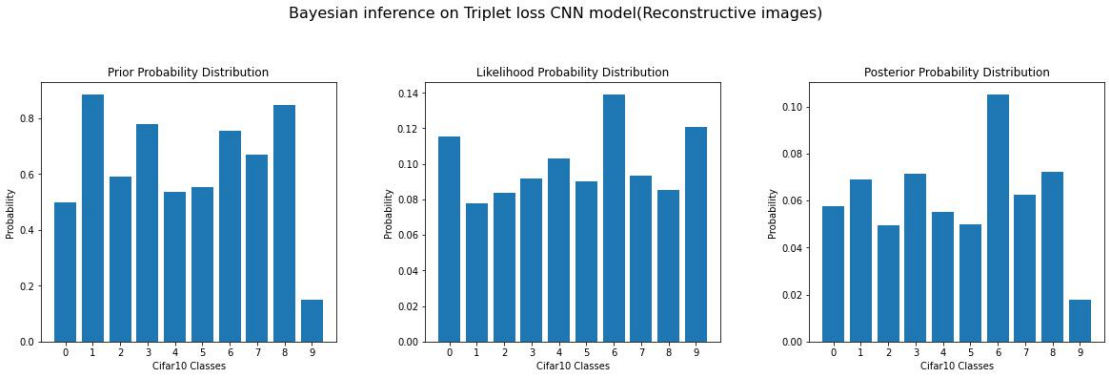


Figure 25. Bayesian inference on Triplet loss CNN model(Reconstructive Images)

After evaluating the models, we picked the higher accuracy rate model that is trained on 30 Epochs. To further analyze this model, we applied Bayesian inference and KL divergence. To begin, we calculated the prior probabilities, which represented our initial beliefs about the likelihood of each class in the Reconstructive CIFAR-10 dataset. Using the trained CNN model, we then calculated the likelihood probabilities, which represented the predicted probabilities for each class. Finally, we used Bayes' theorem to calculate the posterior probabilities, which updated our beliefs about the probability of each class based on the predicted probabilities from the model as shown in [Fig25]. We then computed the KL divergence between the prior and posterior distributions using the posterior probabilities. The KL divergence for this model was 0.5719688499500282 nats. This value suggests that the model's predicted probabilities were not very close to our prior beliefs, indicating that the model may have overfitted to the training data.

5.3 Quadruplet loss Lensless images

Hyperparameters for the following experiment are as follows: Image size = [32]; Learning rate = [0.001]; Optimizer = [Adam]; Batch size = [64]; Quadruplet loss Margin 1 = [0.005]; Quadruplet loss Margin 2 = [0.007]; Activation function = ['relu']; Batch normalization = False; Data augmentation = Random horizontal flip

15 Epochs:

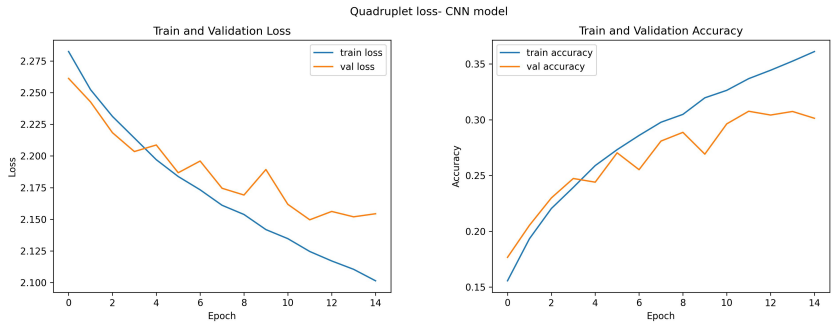


Figure 26. A CNN model trained with Quadruplet loss on 15 epochs

In the train and validation loss graph[Fig26], both the train and validation losses start with high values and gradually decrease with each epoch. However, there is a noticeable gap between the train and validation losses at the beginning, indicating that the model is overfitted to the training data. This is because the training loss is decreasing faster than the validation loss. From epoch 3 to epoch 5, the gap between the two losses decreases, indicating that the model is generalizing better to unseen data. However, after epoch 5, the gap starts to increase again, indicating that the model is overfitting again to the training data.

In the Train and Validation Accuracy graph[Fig26], both the train and validation accuracies start with low values and gradually increase with each epoch. However, there is a gap between the two accuracies at the beginning, indicating that the model is underfitting to the training data. This is because the model is not complex enough to capture the patterns in the training data, leading to low accuracy. From epoch 3 to epoch 5, the gap between the two accuracies decreases, indicating that the model is learning better representations of the data. However, after epoch 5, the gap starts to increase again, indicating that the model is not generalizing well to unseen data.

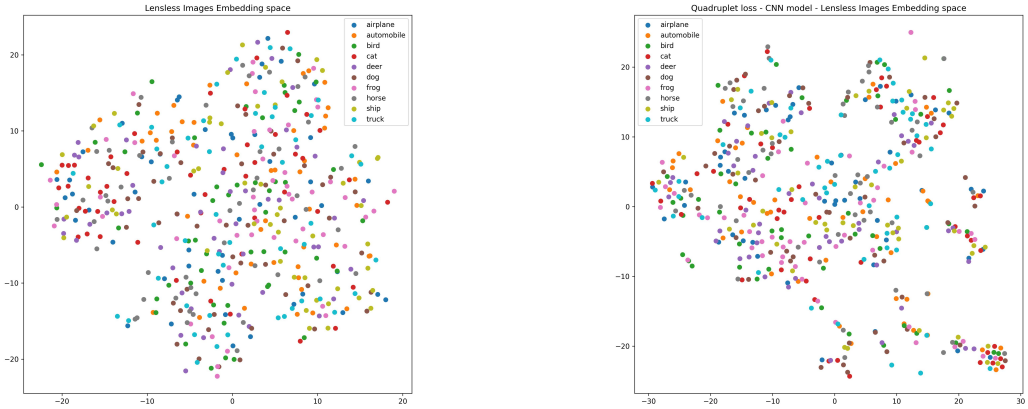


Figure 27. Lensless Images Embedding space at 15 Epochs

When we plot the test images in the embedding space, the model struggles to make clusters of classes[Fig 27] .

30 Epochs:

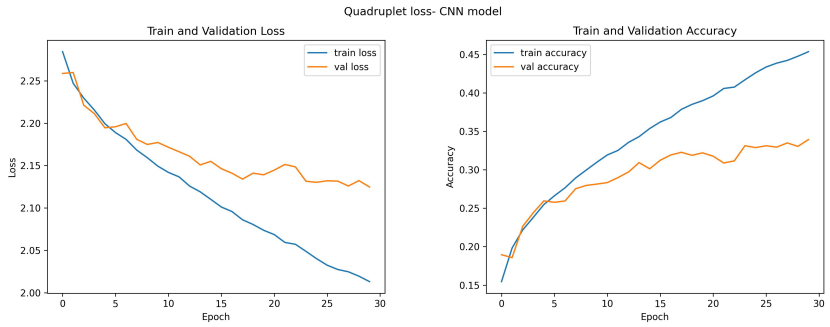


Figure 28. A CNN model trained with Quadruplet loss on 30 epochs

For the train and validation loss graph[Fig28], the model initially experiences a high level of loss in both the train and validation sets. However, as the epochs progress, the loss decreases gradually. Both the train and validation losses start almost from a similar value in the beginning and join until epoch 5. After epoch 5, the gap between the training and validation loss starts to increase, which suggests that the model is starting to overfit the training data. This means that the model is becoming too specific to the training data and may not generalize well to new, unseen data.

For the train and validation accuracy graph[Fig28], the model initially experiences low levels of accuracy, but as the epochs progress, the accuracy gradually increases. The

accuracy starts from a low value in the beginning and joins together until epoch 5. After epoch 5, the gap between the training and validation accuracy starts to increase, which suggests that the model is starting to overfit the training data

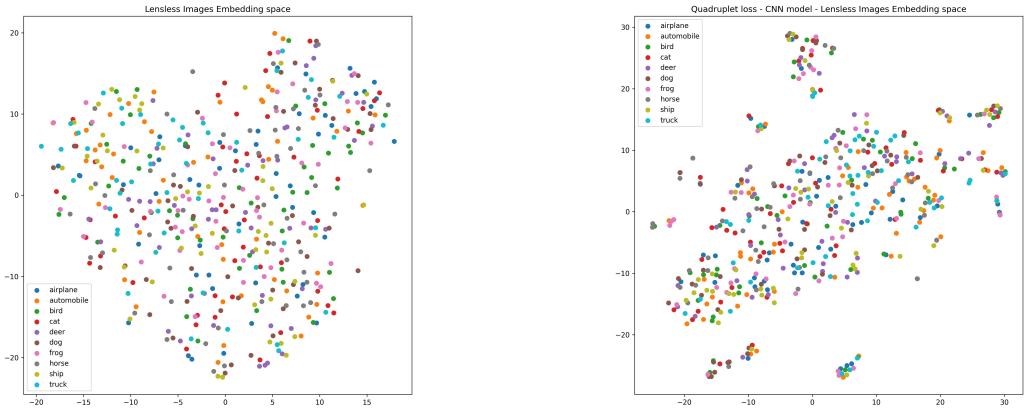


Figure 29. Lensless Images Embedding space at 30 Epochs

When we plot the test images in the embedding space, the model struggles to make clusters of classes [Fig 29].

50 Epochs:

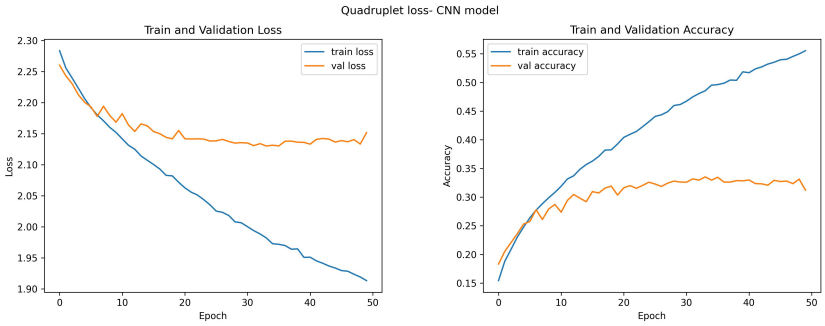


Figure 30. A CNN model trained with Quadruplet loss on 50 Epochs

In the train and validation loss graph[Fig30], the model's performance improves during the first few epochs as both the train and validation losses decrease. However, after about seven epochs, the gap between the training and validation losses starts to increase, indicating that the model is starting to overfit the training data. This means that the model is becoming too complex and is fitting the noise in the training data rather than learning the underlying patterns that generalize to new data.

In the train and validation accuracy graph[Fig30], the model's performance improves over time as both the train and validation accuracies increase. However, the gap between the training and validation accuracies also increases over time, indicating that the model is overfitting the training data. This means that the model is becoming too specialized to the training data and is not able to generalize to new data.

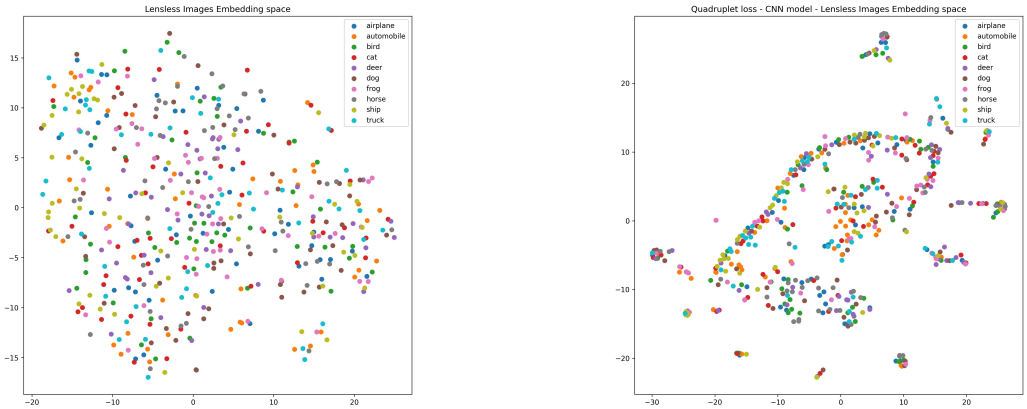


Figure 31. Lensless Images Embedding space at 50 Epochs

When we plot the test images in the embedding space, the model is able to make clusters but struggles in making similar class clusters [Fig31].

Table 3. Quadruplet Loss

Lensless Images	Accuracy	Precision	Recall	F1score	Epochs
test images	30.16	29.78	30.16	29.53	15
test images	31.84	31.57	31.84	31.24	30
test images	32.14	31.85	32.14	31.71	50

Overall, when we give test images to our model to classify lensless Cifar10 images, the model that is trained on 50 epochs scores the highest.

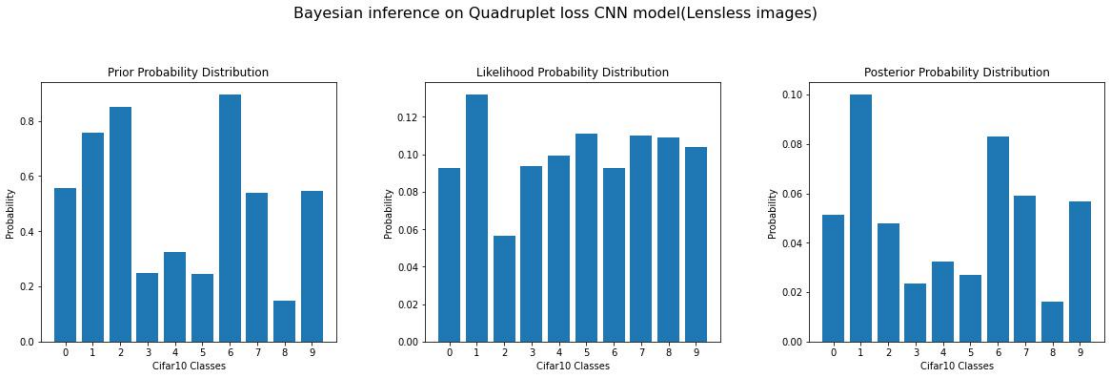


Figure 32. Bayesian inference on Quadruplet loss CNN model(Lensless Image)

After evaluating the models, we picked a higher accuracy model that is trained on 50 Epochs. To gain further insights into this model, we apply Bayesian inference and KL divergence. Bayesian inference allowed us to calculate the posterior probabilities, which represent updated beliefs about the probability of each class based on the predicted probabilities from the model as shown in [Fig32]. We then calculated the KL divergence between the prior and posterior distributions using the posterior probabilities for this model, which was found to be 0.8383306787696252 nats.

The KL divergence value suggests that the model's predicted probabilities were quite different from our prior beliefs, indicating that the model may have overfitted to the training data.

5.4 Quadruplet loss Reconstructive Images

Hyperparameters for the following experiment are as follows: Image size = [32]; Learning rate = [0.001]; Optimizer = [Adam]; Batch size = [64]; Quadruplet loss Margin 1 = [0.005]; Quadruplet loss Margin 2 = [0.007]; Activation function = ['relu']; Batch normalization = True; Data augmentation = Random horizontal flip, Random vertical flip

15 Epochs:

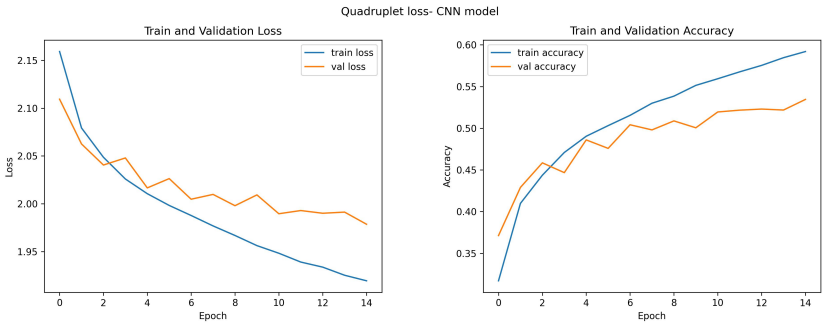


Figure 33. A CNN model trained with Quadruplet loss on 15 epochs

In the train and validation loss graph[Fig33], the model initially struggles to learn the underlying patterns in the data, as evidenced by the high initial losses for both train and validation sets. However, as the model trains over the 15 epochs, we see a gradual decrease in both the train and validation loss. This suggests that the model is getting better at generalizing to new data, as the decrease in validation loss indicates that the model is not overfitting to the training data. However, we do notice that the gap between the training and validation loss starts to increase from epoch 5, indicating that the model may be starting to overfit the training data.

In the train and validation accuracy graph[Fig33], we observe a similar trend to that of the loss graph. The model's accuracy initially starts low for both the train and validation sets but gradually improves over the 15 epochs. The fact that the validation accuracy is consistently lower than the training accuracy suggests that the model is overfitting to some extent. Additionally, the gap between the training and validation accuracy starts to increase from epoch 4, indicating that the model's performance on the training data is improving at a faster rate than its performance on the validation data.

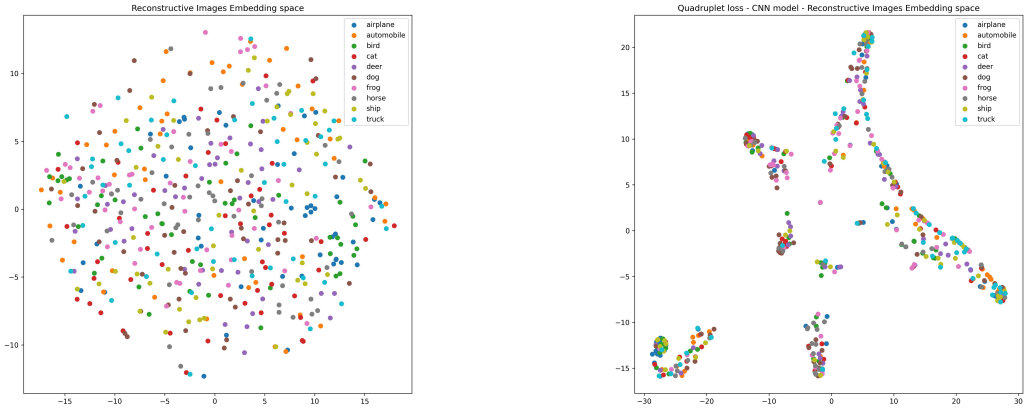


Figure 34. Reconstructive Images Embedding space at 15 Epochs

When we plot the test images in the embedding space, the model is able to make similar class clusters but struggles to make them properly[Fig 34].

30 Epochs:

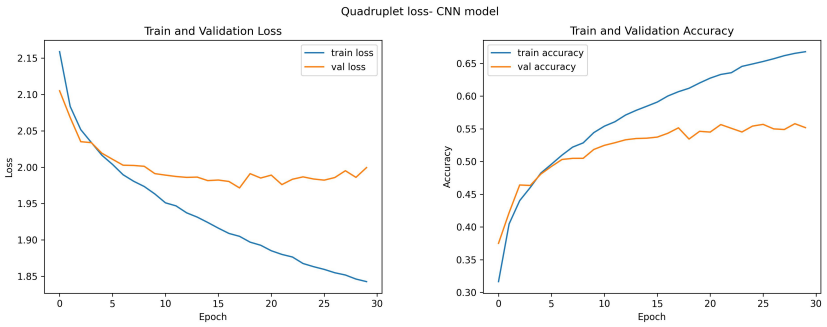


Figure 35. A CNN model trained with Quadruplet loss on 30 epochs

From the train and validation loss graph[Fig35], it is evident that the model is initially overfitting as both the train and validation loss are close together, but with increasing epochs, the gap between them widens. This suggests that the model is learning to fit the training data too well and is not generalizing well to the validation data.

From the train and validation accuracy graph[Fig35], it is clear that the model is learning well, as both the train and validation accuracy increase with each epoch. However, as with the loss graphs, the gap between the train and validation accuracy widens with increasing epochs. This indicates that the model is becoming more specialized in predicting the training data but is not generalizing well to new data. It is worth noting that

the final values of the training and validation accuracy suggest that the model performs better on the training data than on the validation data.

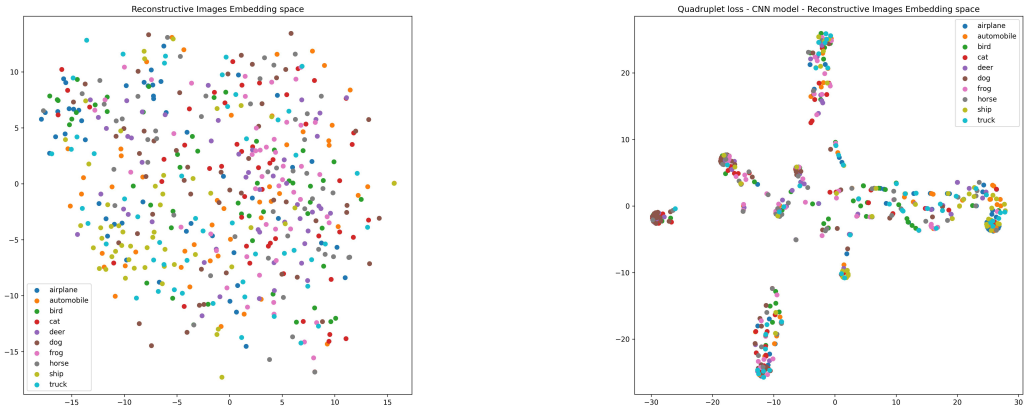


Figure 36. Reconstructive Images Embedding space at 30 Epochs

When we plot the test images in the embedding space, the model is able to make similar class clusters but struggles to make them properly [Fig 36].

50 Epochs:

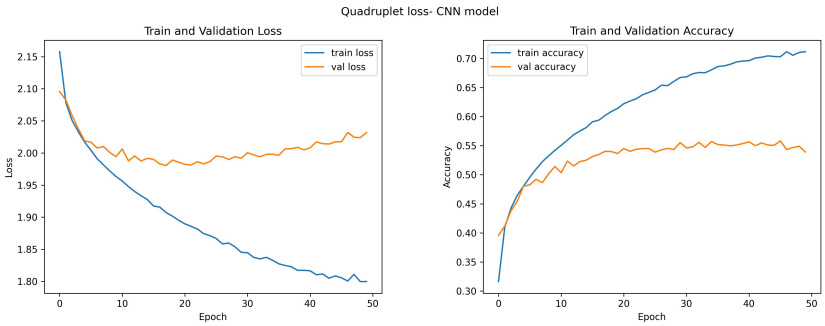


Figure 37. A CNN model trained with Quadruplet loss on 50 epochs

In the train and validation loss graph[Fig37], the train and validation loss start high and gradually decreases as the epochs number increases. This indicates that the model is learning from the data and improving its performance. However, the gap between the training and validation loss starts to increase after epoch 5, which suggests that the model is overfitting to the training data. This means that the model is becoming too complex and is fitting the noise in the training data rather than learning the underlying patterns.

In the Train and Validation Accuracy graph[Fig37], it is observed that both the train and validation accuracy start low and gradually increase with the increase in the number of epochs. This indicates that the model is learning and improving its performance. However, the gap between the training and validation accuracy also starts to increase after epoch 5, which suggests that the model is overfitting to the training data. This means that the model is becoming too specialized in predicting the training data and is not generalizing well to new data.

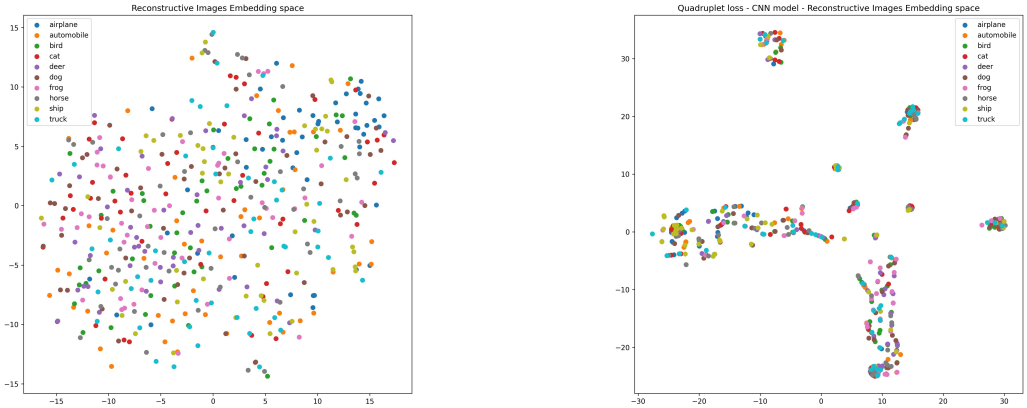


Figure 38. Reconstructive Images Embedding space at 50 Epochs

When we plot the test images in the embedding space, the model struggles to make class clusters properly[Fig 38].

Table 4. Quadruplet Loss

Reconstructive Images	Accuracy	Precision	Recall	F1score	Epochs
test images	40.76	40.44	40.76	40.26	15
test images	40.78	40.51	40.78	40.44	30
test images	40.71	41.26	40.71	40.47	50

Overall, when we give test images to our model to classify reconstructive Cifar10 images, the model that is trained on 30 epochs scores the highest.

By applying causality to Quadruplet loss, we achieve an increase in accuracy and recall by 0.10; precision by 0.48; and F1-score by 0.39 in the 30 Epoch's model.

Table 5. Quadruplet Loss with Causality

Reconstructive Images	Accuracy	Precision	Recall	F1score	Epochs
test images	39.57	39.58	39.57	38.91	15
test images	40.88	40.99	40.88	40.83	30
test images	40.50	41.42	40.50	40.26	50

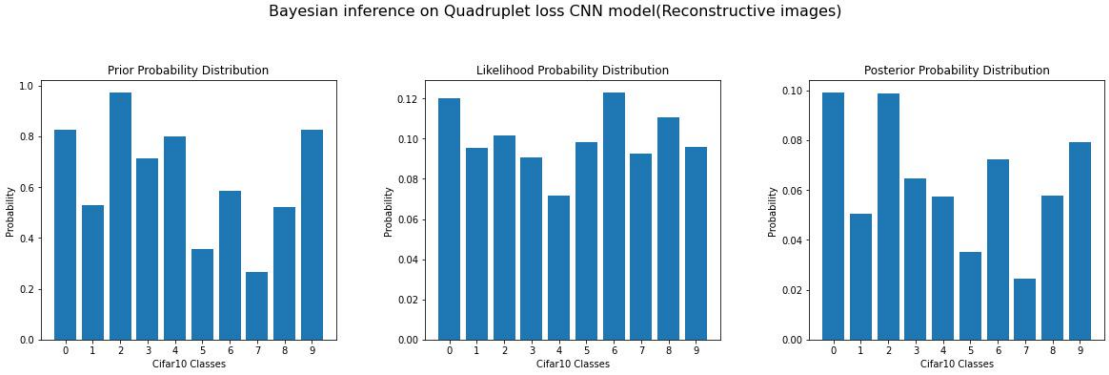


Figure 39. Bayesian inference on Quadruplet loss CNN model(Reconstructive Image)

To further analyze the high-accuracy models that showed promising results and trained on 30 Epochs, we applied Bayesian inference and KL divergence. After calculating the prior, likelihood, and posterior probabilities[Fig39], we used Bayes' theorem to calculate the KL divergence between the prior and posterior distributions. The KL divergence for this model was 0.5249983517897225 nats. This value suggests that the model's predicted probabilities were somewhat close to our prior beliefs, indicating that the model may have some generalization ability.

6 Conclusion

This thesis proposed a novel approach to solve the problem of lensless image classification. The author trained a CNN model on a paired dataset consisting of two pairs: The first pair included the Lensless and Original Cifar10 datasets, while the second pair comprised the Reconstructive and Original Cifar10 datasets. For image classification, the author employed deep metric learning methods such as triplet loss and quadruplet loss with cross-entropy. The proposed approach uses a similarity metric called Euclidean distance as its learning algorithm, enabling fast and efficient classification without the need for image reconstruction methods. The experiments demonstrate the proposed approach's effectiveness in achieving efficient classification of lensless images. Moreover, the approach has the potential to impact the development of camera technology in the future by making cameras more lightweight and lens-free and can potentially be used in various fields, such as biomedical and surveillance. One of the key advantages of the proposed approach is that it is less time-consuming than other approaches, where image reconstruction is required before the classification of lensless images. The proposed approach directly classifies the lensless images, eliminating the need for image reconstruction, which can be a time-consuming process. The future work related to the approach would be accumulating data pairs that consist of lensless and high-quality counterpart images. This would help to improve the performance of the model by providing additional information to aid in the classification process. Additionally, working on increasing the resolution or denoising of the original Cifar10 dataset, or working on textual annotation could also be considered to further improve the model's performance. Overall, the proposed approach is a promising solution to the problem of lensless image classification, and further research in this area could lead to significant improvements in the field of optics-free image classification for real-world applications. The findings of this thesis highlight the importance of lensless camera and the potential it holds for future applications. The advantages of the proposed approach, such as its efficiency and time-saving capabilities, make it a practical solution for real-world applications, and its potential impact on the field of lensless image classification is significant.

References

[AL20] Brandon Amos Aaron Lou, Maximilian Nickel. Deep riemannian manifold learning, December 11, 2020.

[B19] Harikrishnan N B. Evaluate the performance of a machine learning model, Dec 10, 2019.

[BARV20] Vivek Boominathan, Jesse K. Adams, Jacob T. Robinson, and Ashok Veeraraghavan. Phlatcam: Designed phase-mask based thin lensless camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(7):1618–1629, 2020.

[Ber21] Etienne Bernard. Bayesian inference, introduction to machine learning, 2021.

[BVS22] Eric Bezzam, Martin Vetterli, and Matthieu Simeoni. Learning rich optical embeddings for privacy-preserving lensless image classification, 06 2022.

[CCZH17] Weihua Chen, Xiaotang Chen, Jianguo Zhang, and Kaiqi Huang. Beyond triplet loss: A deep quadruplet network for person re-identification. 07 2017.

[CGZ+16] De Cheng, Yihong Gong, Sanping Zhou, Jinjun Wang, and Nanning Zheng. Person re-identification by multi-channel parts-based cnn with improved triplet loss function. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1335–1344, 2016.

[Den12] Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[GRLGPC20] Elliott Gordon Rodriguez, Gabriel Loaiza-Ganem, Geoff Pleiss, and John Cunningham. Uses and abuses of the cross-entropy loss: Case studies in modern deep learning, 11 2020.

[HBL17] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. 03 2017.

[KKPM17] Ganghun Kim, Stefan Kapetanovic, Rachael Palmer, and Rajesh Menon. Lensless-camera based machine learning for image classification. 09 2017.

[KNH] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research).

[KSB+20] Salman Khan, Varun Sundar, Vivek Boominathan, Ashok Veeraraghavan, and Kaushik Mitra. Flatnet: Towards photorealistic scene reconstruction from lensless measurements. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP:1–1, 10 2020.

[MMBJ+21] Monireh Mohebbi Moghaddam, Bahar Boroomand, Mohammad Jalali, Arman Zareian, Alireza Daeijavad, and Mohammad Hossein Manshaei. Game of gans: Game theoretical models for generative adversarial networks, 06 2021.

[MYK+19] Kristina Monakhova, Joshua Yurtsever, Grace Kuo, Nick Antipa, Kyrollos Yanny, and Laura Waller. Learned reconstructions for practical mask-based lensless imaging. *Optics Express*, 27:28075, 09 2019.

[NM22a] Soren Nelson and Rajesh Menon. Bijective-constrained cycle-consistent deep learning for optics-free imaging and classification. *Optica*, 9(1):26–31, Jan 2022.

[NM22b] Soren Nelson and Rajesh Menon. Bijective-constrained cycle-consistent deep learning for optics-free imaging and classification. *Optica*, 9(1):26–31, Jan 2022.

[ON15] Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks. *ArXiv e-prints*, 11 2015.

[POW+10] Karthir Prabhakar, Sangmin Oh, Ping Wang, Gregory D. Abowd, and James M. Rehg. Temporal causality for the analysis of visual events. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1967–1974, 2010.

[RKJ21] Joshua Rego, Karthik Kulkarni, and Suren Jayasuriya. Robust lensless image reconstruction via psf estimation. pages 403–412, 01 2021.

[TNA+19] Jasper Tan, Li Niu, Jesse K. Adams, Vivek Boominathan, Jacob T. Robinson, Richard G. Baraniuk, and Ashok Veeraraghavan. Face detection and verification using lensless cameras. *IEEE Transactions on Computational Imaging*, 5(2):180–194, 2019.

[Wik] Wikipedia. Euclidean distance.

[WY16] Hao Wang and Dit-Yan Yeung. A survey on bayesian deep learning, 2016.

[Zac21] Zach. How to calculate kl divergence in python, December 6, 2021.

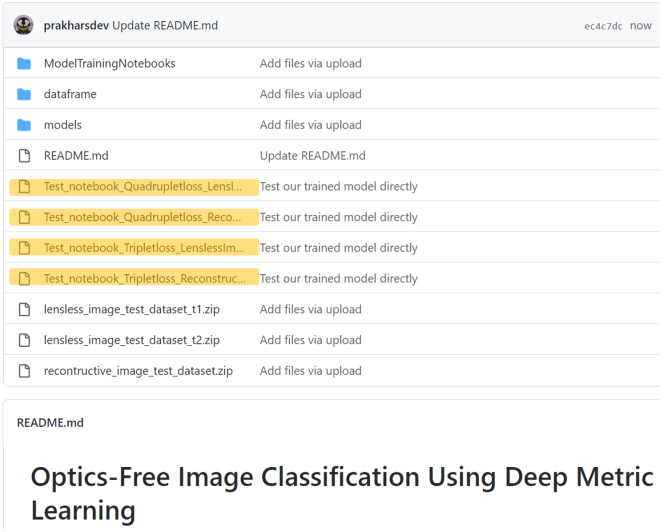
[ZLT+22] Xianquan Zhang, Xuelong Li, Zhenjun Tang, Shichao Zhang, and Shaomin Xie. Noise removal in embedded image with bit approximation. *IEEE Transactions on Knowledge and Data Engineering*, 34(3):1359–1369, 2022.

Appendix

I. Access to the code

https://github.com/prakharsdev/Optics_Free_ImageClassification

Note: Our model(s) can be tested by directly downloading notebooks with names that begin with "Test_notebook_" as shown below:



II. Licence

Non-exclusive licence to reproduce thesis and make thesis public

I, **Prakhar Srivastava**,
*(*author's name)

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to

   reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

   Optics-free Image Classification with Deep Metric Learning,
   *(*title of thesis)

   supervised by Dr. Kallol Roy.
   *(*supervisor's name)

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Prakhar Srivastava
09/03/2023