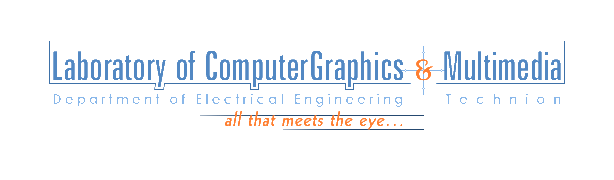
Text

Description automatically generated



**044167 Project A – Project Portfolio**

|  |  |
| --- | --- |
| Computer Graphics and Multimedia | **Lab Name** |
| Backward Compatibility in Face Recognition | **Project Name** |
| 6461 | **Project ID** |
| Winter 2021/2022 | **Semester** |
| Elad Hirsch | **Supervisor** |
| Shai Yehezkel 205917883  Lior Dvir 207334376 | **Students** |
|  |

Table of Contents

[Abstract 2](#_Toc102424601)

[Background 3](#_Toc102424602)

[Machine Learning 3](#_Toc102424603)

[Neural Network 3](#_Toc102424604)

[Model Architecture 4](#_Toc102424605)

[4](#_Toc102424606)

[Embedding Vector, Embedding Space, Manifold 4](#_Toc102424607)

[5](#_Toc102424608)

[Recognition and Retrieval 5](#_Toc102424609)

[Project Definition 6](#_Toc102424610)

[Cross Model Compatibility 6](#_Toc102424611)

[Naïve Approach 8](#_Toc102424612)

[Unified Transformer 9](#_Toc102424613)

[Embedding Space Transformer 10](#_Toc102424614)

[Proposed Improvements 10](#_Toc102424615)

[Assumptions 11](#_Toc102424616)

[Challenges 11](#_Toc102424617)

# Abstract

# Background

To get familiar with notations and symbols we use in our work, this section is dedicated for going through preliminaries and concepts that will be in our project.

## Machine Learning

The study of computer [algorithms](https://en.wikipedia.org/wiki/Algorithm) that can improve automatically through experience and using data[[1]](#footnote-2).

In our scope data will be images of human faces. At some point we will be handling embedding vectors, a type of data which is extracted from images.

Our training is being conducted in a **Supervised Learning** fashion, E.G our input data is always labeled as we know what the desired output is.

## Neural Network

One of many ways to solve optimization problems is with the use of parametrized models. A state-of-the-art model is a neural network which is widely used in computer vision and machine learning tasks.

A neural network has an output and possibly multiple outputs.



* Female
* Has Glasses
* Light ginger hair color

Figure 1 - Neural Network

## Model Architecture

Neural networks differ in the number of layers, dimensions of each layer, activation functions and loss functions. The set of these define a specific model architecture.



Figure 2 - Model Architectures

Note that when a transformer is addressed in this portfolio, we do not regard to the architecture family of transformers with self-attention mechanisms, but rather a model which is fed an input, and outputs data with manipulations done within.

## Embedding Vector, Embedding Space, Manifold

An embedding vector is the output of the last layer of a neural network **before** performing any task (E.G classification). The vector represents deep features and is of can be of any size desired (as a model’s parameter).



NN

Embedding Vector

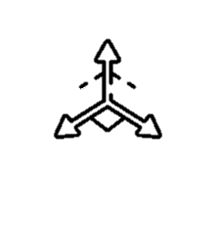
Figure 3- Embedding Vector

In our project we define the embedding vector size to be 512, that is each RGB image will be represented a deep feature vector of size 512.

When looking at multiple images, we can imagine a multidimensional space spanned by the multiple embeddings, each correlate to a single point in that space. In literature it might be called Deep Feature Space.



NN



Embedding Space

Figure 4 - Embedding Space

Note every network might have a different embedding space.

As this space is not fully sampled, we can imagine a hyper-surface representing out samples in the embedding space called a manifold. Every sample from the dataset has a corresponding representation in the dimension space sampled in the manifold.

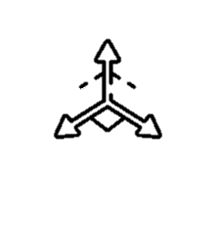


Figure 5 - Manifold

## Recognition and Retrieval

As said in previous section, a dataset of images will span an embedding space with specific neural network. The space can be saved into a gallery of known images of people.

As far as privacy issues concern, the gallery consists of embedding vectors, and not of images.

When a specific image is fed-forward through the network, we can define the process of querying the gallery with an embedding vector as retrieval.

A vector is associated with a unique person, and thus a recognition process is achieved by finding the closest match to it in the gallery and returning the label associated.

Gallery Embeddings

Search

(Nearest Neighbor)



NN

Embedding Vector

Alice

* Person is Alice

Figure 6 - Recognition

# Project Definition

Our work is based on the following paper: [Unified Representation Learning for Cross Model Compatibility](https://arxiv.org/abs/2008.04821)

This section is dedicated for explaining what the Cross Model Compatibility problem is, what is our project scope, and what we propose for enhancing existing approaches.

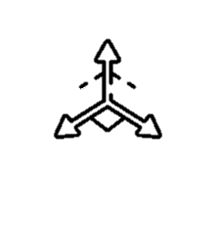
## Cross Model Compatibility

Visual recognition and retrieval systems are widely used in out lives. Examples for use are frictionless physical access systems such in Apple ID, missing persons search, global place recognition and even homeland security applications.

et us recall the standard process of recognizing a person – that is feeding forward an image through the network, receiving its embedding vector representation, and with that vector query an existing gallery of embeddings associated to a specific network. The label of the closest match the embedding vector will be the person’s ID.

Let:

* be a neural network which extracts deep features
* the spanned embedding space by .



NN #1

Embedding Space A

Gallery Embeddings

Figure 7 - Standard Retrieval System

As research progresses, more deep network architectures are discovered, ones with potentially a better performance than . Let be a new architecture which improves our current retrieval system, in the sense that the percentage of people that are recognized (or not recognized when they do not exist) is higher.

Recall each network spans a different embedding space which might differ both in dimensions and the manifold of our samples.

NN #1

Embedding Space A

Gallery Embeddings

Search (Nearest Neighbor)

NN #2

Embedding Space B

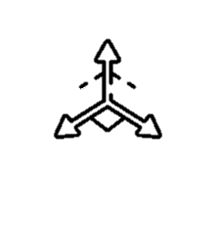
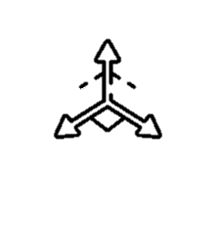


Figure 8 - Second NN

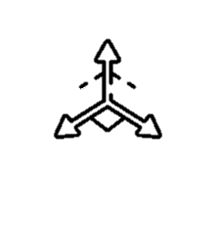
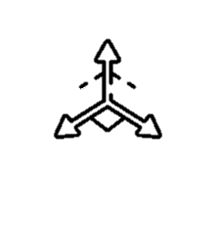
We would like now to transfer our work to the new architecture, so that every image now will be fed to . However, querying directly from the old gallery embeddings will probably yield low accuracies as the manifold differs.

Addressing how a new neural network with a new embedding space can retrieve valid information from the old gallery embeddings is what is called the Cross Model Compatibility.

## Naïve Approach

A naïve approach to solve this problem will be taking the entire dataset and feeding it through , and create a new gallery embedding.

Figure 9 - Naive Approach



NN #1

Embedding Space A

Gallery Embeddings A

Search (Nearest Neighbor)

NN #2

Embedding Space B



Gallery Embeddings B

Search (Nearest Neighbor)

However, this approach will require re-embedding all existing images of identified people. This process can take a long time, and moreover – it imposes privacy issues as this will require saving images of people, and not only their embedding vectors.

## Unified Transformer

A thorough work proposed by Wang et al. involves creating a unified transformer which is trained together but transforms independently and Embedding Space into and embedding space .

NN #1

Embedding Space A

Gallery Embeddings A

NN #2

Embedding Space B



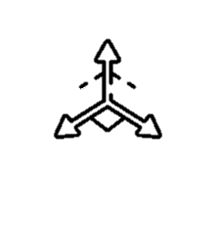
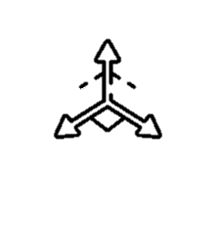
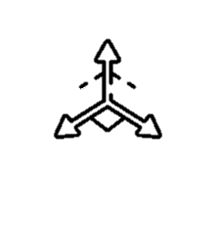
Embedding Space A'



Gallery Embeddings B

Search

Search



Unified Transformer

Figure 10 - Unified Transformer

This approach optimizes a new and unified embedding space and leverages information both from the original gallery and new embedding space . In our work decided to focus on a different approach also reviewed in the mentioned paper.

## Embedding Space Transformer

Another approach which might yield satisfactory results would be training a transformer which transforms embedding space into the gallery embedding space.

NN #1

Embedding Space A

Gallery

Embeddings A

Search (Nearest Neighbor)

NN #2

Embedding Space B



Embedding Space A'



Transformer

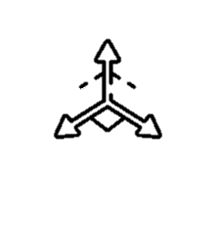
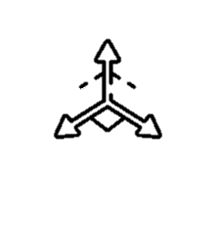
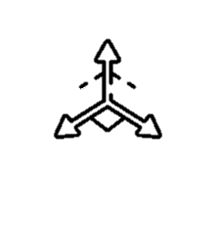


Figure 11 - Space Transformer

In our work we implemented this solution and added enhancements which improved the accuracy of recognition after feeding .

## Proposed Improvements

To the existing transformer we will add new loss functions to the optimization process.

Moreover, we will perform interpolation of embedding vectors resulting in new samples.

A good interpolation is a one which will create new samples which are on the manifold of , thus introducing new information into the system.

Interpolation methods and loss functions will be discussed in next sections.

Figure 12 - Interpolation



## Assumptions

To simplify our task and to adapt the Cross Model Compatibility problem to the time scope of out project, a few basic assumptions were made and were taken into consideration in our work.

Our work is based on recognition systems in the [InsightFace](https://github.com/deepinsight/insightface) repository.

The dataset we are working with is [*MS1M-ArcFace*](https://drive.google.com/file/d/1SXS4-Am3bsKSK615qbYdbA_FMVh3sAvR/view) which holds 5.8 million RGB images of almost 90,000 different people (labels), where In this dataset, all images are visible, centered and aligned.

Moreover, we will assume the networks that will serve as and will perfectly label all existing images in the dataset, as we do not consider any loss of performance incurred by miss-classifying them. This assumption turns to be quite accurate as we trained two backbones with accuracies of 96%-98% in labeling images.

Lastly, we do not limit the number of parameters used by our transformer as this allows us to explore different architectures and focusing on the improvements we introduced.

## Challenges

Before starting our project, we tried identifying some of the possible difficulties we might face. These include:

Adjustments of InsightFace repository – The repository is large and consists of 2M+ lines of code. Finding the correct modules that we will benefit from and changing some of its code in the embedding vector manipulation.

Computation Limitations – As we process a large amount of data, and we are pre-processing many embedding vectors before even training our transformer. Moreover, training network, both InsightFace’s networks and our transformer might take a long time.

Limited Timeframe – As this is our project for submission, the extent in which we will be able to improve our transformer is limited.

1. <https://en.wikipedia.org/wiki/Machine_learning> [↑](#footnote-ref-2)