

---

# Project Title: Exploring Autoencoders with Simple Images of Shapes

---

**Kevin Lewis**  
lewis.3164@osu.edu

**Michael Lin**  
lin.3976@osu.edu

**Shashank Raghuraj**  
raghuraj.2@osu.edu

## 1 Project Type

Reproducing examples in the textbook

## 2 Project Introduction

Please give **an introduction (1-2 paragraphs)** about your project. For example, what CVPR competition you will work on, what algorithm you are going to reproduce, what set of algorithms you are going to benchmark and on which dataset, which parts of the textbook you are going to reproduce, etc.

Please include URL and/or reference so that I can quickly find out the corresponding references and sources. If you want to add a reference, go to Google Scholar, search for the corresponding papers, click on cite, and copy the text in BibTex into the ref.bib file. Finally, use [? ] to add the reference into your text. If you want to add a URL, do <https://dinov2.metademolab.com/>.

In this project, we will replicate the textbook example on autoencoders, as seen here:

[https://visionbook.mit.edu/representation\\_learning.html#experiment-do-autoencoders-learn-useful-representations](https://visionbook.mit.edu/representation_learning.html#experiment-do-autoencoders-learn-useful-representations)

## 3 Project Motivation

This project is interesting for a couple of reasons. First, the model is not too difficult itself, so we may be able to create this autoencoder without using any machine learning libraries. This will be a useful learning tool for understanding more about how these models work behind the scenes. Also, exploring the different things that are possible with the encoded representations (for example, animating a transition between different shapes).

## 4 Project Plan

### 4.1 Baseline approach

We will use a simple fully connected neural network as a baseline approach for our project. This is because it should be enough for our purposes, and it is simple enough that we will be able to build it ground up. One potential limitation of this is that using a fully connected network will limit the sizes of the hidden layers. If they were too big, the model may be too big and slow to train effectively.

### 4.2 Advanced approach

Beyond this, we may add some convolutional layers to see how they impact the performance and accuracy of the model. This will allow us to have a few layers without needing to decrease layer size.

We may also add some skip connections in these early layers to emulate a ResNet, in order to explore how this change affects accuracy.

#### **4.3 Validation plan**

We will have a dataset of shapes with different colors, sizes, and rotations to test our network on. We do not currently have access to this dataset, but it will be easy to generate a dataset like it. The textbook used 64,000 images, so we will look to generate a dataset of similar size.

#### **4.4 Computational resources**

This model will be simple enough to run on our own computers.

#### **4.5 Library, Tool, etc.**

We will write this in Python. The main library we use will likely be either Numpy or CuPy for fast matrix operations, though we will likely write the actual neural network code ourselves.

#### **4.6 Estimated baseline algorithm runtime**

We are generating the dataset ourselves, and since the dataset the textbook used is not too big, we can tune the size of our images so that one epoch worth of training takes about a minute.

### **5 Workload**

As mentioned in the class, the final project is 30% of your overall grade so it must contain sufficient workload (i.e., not something you can simply complete in several hours or a day.) The workload may include reading papers, get familiar with tools, organize data, implement algorithms, trial and errors, think about new ideas, etc.

Please note that you are welcome (and encouraged) to leverage any existing code and/or implementation, either as the baseline or as the building blocks of your advanced approach. It makes little sense if you keep yourself blind of the existing resources that you can leverage.

### **6 Ideal result**

If everything is successful, what will be the outcome and result of your project, for example, achieving the best performance on CVPR 2025 competition.

#### **6.1 Insights**

If you successfully complete your final project, what will be the insights that you will gain?

### **7 Potential Risk**

Any potential risks that you may not complete your final project or your approach may not work? Please envision what may prevent you from completing the project. These may include your designed algorithm will not work as you expect, insufficient computational resources, etc.

### **8 Duplication Statement**

If you choose a self-defined project, it is okay if it is related to your own research project (e.g., at your lab). However, you **MUST** write a short paragraph clearly articulating the difference between your final project and your own research project. More specifically, it is **NOT** allowed if you bring what you already planned to do in your own research project (before taking this class) as the final project.

If you plan to reproduce an existing algorithm, please clearly specify whether there are accessible code/implementation online. If so, please clearly state how your project will not just be copying the code and rerunning it.