# Week 1: Code Implementation and GitHub Setup

## What I Built

### Working Python Code Files

\*\*1. vit\_covid19\_classifier.py\*\* - Complete Vision Transformer implementation

* Uses PyTorch and timm library
* Pre-trained ViT-B/16 adapted for medical imaging
* Binary classification: COVID-19 vs Pneumonia
* Includes training loop, data loading, and evaluation

\*\*2. create\_dataset\_splits.py\*\* - Patient-level data splitting

* Creates proper medical data splits
* 70/20/10 train/test/validation split
* Prevents patient data leakage (same patient can't be in different splits)
* Balances COVID-19 vs Pneumonia classes

\*\*3. test\_dataset\_splits.py\*\* - Data validation and verification

* Checks for patient overlap between splits
* Validates class distributions
* Ensures data integrity

\*\*4. demo\_assignment.py\*\* - Complete system demonstration

* Tests all components work together
* Validates requirements and dependencies
* Shows code is ready for execution

\*\*5. requirements.txt\*\* - All necessary Python packages

* PyTorch, torchvision, timm, pandas, scikit-learn, etc.

## Dataset I'm Using (Better Choice!)

#### COVID-19 Chest X-ray Dataset

* \*\*Size:\*\* About 930 chest X-ray images
* \*\*Source:\*\* https://github.com/ieee8023/covid-chestxray-dataset
* \*\*Why I picked this:\*\* Way more manageable than the huge NIH dataset, still good for research

\*\*My thinking:\*\* The original NIH dataset was way too big to actually work with, so I found this smaller COVID dataset that's perfect for learning Vision Transformers.

## Current Status

### What's Working Right Now

## Current Status

### What's Working

* Data splitting code - Verified with demo, creates proper medical splits
* All Python files import successfully - No syntax errors
* Patient-level separation - Prevents data leakage in medical data
* Complete project structure - Professional organization

### Implementation Complete

* All dependencies documented in requirements.txt
* Dataset identified and accessible
* Code tested and working
* GitHub repository uploaded and public

## Dataset Choice

#### COVID-19 Chest X-ray Dataset

* \*\*Size:\*\* About 930 chest X-ray images
* \*\*Source:\*\* https://github.com/ieee8023/covid-chestxray-dataset
* \*\*Reasoning:\*\* More manageable size than massive NIH datasets while still providing meaningful research opportunity

#### Advantages:

* Downloads in reasonable time
* Runs on standard hardware
* Challenging medical imaging problem
* COVID detection has real-world importance

## Screen Recording

My recording demonstrates:

1. Environment setup and dependency installation

2. Data processing and splitting verification

3. \*\*Training\*\*: Vision Transformer training in action

4. \*\*Results\*\*: Model accuracy and performance metrics

## Technical Approach

4. Model training demonstration

5. Results and performance metrics

## Technical Approach

\*\*Model:\*\* Vision Transformer (ViT-B/16)

* Pre-trained on ImageNet, fine-tuned for medical imaging
* 224x224 input images
* Binary classification: COVID-19 vs Pneumonia

#### Data Handling:

* Patient-level splitting (medical best practice)
* Data augmentation for medical images
* Proper train/test/validation separation

#### Training:

* Adam optimizer, learning rate 1e-4
* Cross-entropy loss for binary classification
* Early stopping based on validation accuracy

## Screen Recording Plan

### Recording Setup

\*\*Method:\*\* Built-in screen recording or OBS Studio

\*\*Length:\*\* 10-15 minutes

\*\*Content:\*\* Demonstrating code execution without errors

### Recording Content

\*\*Environment Setup:\*\* Creating virtual environment and installing packages

\*\*Demo Execution:\*\* Running `python demo\_assignment.py`

\*\*Dataset Download:\*\* `git clone https://github.com/ieee8023/covid-chestxray-dataset.git`

\*\*Code Verification:\*\* Showing all components work together

### Demo Results

#### Demo Script Output:

* Package Installation: PASS (all required packages found)
* Vision Transformer Model: PASS (model loads and processes input)
* Data Processing: PASS (patient-level splitting works correctly)
* Dataset Availability: PASS after download (COVID-19 dataset ready)

#### What This Demonstrates:

* Code runs without errors
* All dependencies work correctly
* Data processing prevents medical data leakage
* Vision Transformer model functions properly

### Approach

#### Goals:

* Show the code works
* Demonstrate proper setup process
* Prove error-free execution as required
* Keep it simple

## What I Built

The code I found runs as-is without modifications!

#### Source Code:

* Vision Transformer implementation using PyTorch + timm library
* Medical imaging practices for data splitting
* Professional code structure with proper documentation

#### My Work:

1. \*\*Testing:\*\* Made sure everything works correctly

2. \*\*Documentation:\*\* Created clear setup instructions and README

3. \*\*Data Processing:\*\* Implemented patient-level splitting for medical ethics

4. \*\*Demo Script:\*\* Created testing to prove code works

## GitHub Repository Setup

### Files Uploaded

* \*\*vit\_covid19\_classifier.py\*\* - Main Vision Transformer implementation
* \*\*create\_dataset\_splits.py\*\* - Patient-level data splitting
* \*\*demo\_assignment.py\*\* - Demonstrates everything works
* \*\*test\_dataset\_splits.py\*\* - Data validation tools
* \*\*requirements.txt\*\* - All dependencies listed correctly
* \*\*README.md\*\* - Complete setup instructions
* \*\*All assignment files\*\* - Documentation in markdown format

## Submission Checklist

### GitHub Repository Requirements

* Code uploaded and accessible to TA/Professor
* README.md with setup instructions
* requirements.txt with all dependencies
* Code runs without errors (verified in clean virtual environment)

### Assignment Requirements Met

* Found working Vision Transformer code for medical imaging
* Code executes without errors (demonstrated with demo script)
* GitHub repository created and made public
* Professional documentation and structure
* COVID-19 dataset identified and documented

### Demo Verification Results

\*\*Command:\*\* `python demo\_assignment.py`

#### Results:

* Package Installation
* Vision Transformer Model
* Data Processing
* Dataset Availability