COPS Summer of Code 2025

Intelligence Guild

Club Of Programmers, IIT (BHU) Varanasi

CV Track: Convolutional Neural Networks

2-9 June 2025

Official IG Website: https://cops-iitbhu.github.io/IG-website/

All deadlines are strict. No extensions will be granted.

Introduction

COPS Summer of Code (CSOC) is a flagship initiative under the Club Of Programmers, IIT (BHU) Varanasi, with all verticals contributing through focused tracks. This document outlines the prerequisites for the Intelligence Guild vertical.

Modules will be released weekly and from time to time. **Adhere strictly to deadlines**. Submissions will be evaluated on approach, technical correctness, and clarity. The most technically accurate solution may not necessarily be the one chosen; clarity of thought and a well-reasoned approach will be valued more.

Communities

All communication for the programme will be conducted strictly via Discord. Do not reach out through other channels. Resources and updates will be posted on Github, and all notifications will be made via Discord.

Final Report

A concise report may be submitted along with your final assignment. While **not mandatory**, it may strengthen your overall evaluation. Reports must be written in LaTeX and submitted in PDF format only. We are not interested in surface-level descriptions — focus strictly on your analysis, approach, and reasoning. The report itself constitutes the final assignment. No additional files are to be submitted. Refer to the Assignment section for details. Submit your report here.

Contact Details

In case of any doubts, clarifications, or guidance, you can contact one of us. We request that you stick to Discord as the preferred mode of communication for all the questions that you have as it will also benefit others. However, you can reach out to us through other means in case we fail to respond on Discord.

- Manas Agrawal - 9996584770
- Vivek Kumar - 9873432572
- Manav Kumar Jalan - 9395002919

Resources

Let us begin by gaining an introductory understanding of computer vision through the following blog: Introduction to Computer Vision.

In this module, we will delve into the fundamentals of image processing and Convolutional Neural Networks (CNNs). Additionally, we will explore the basics of various downstream tasks, with a particular focus on image classification.

Image Preprocessing

This section provides useful resources for learning the basics of image preprocessing:

- **Learning Resources:**
  - **CampusX:** CampusX offers a comprehensive and beginner-friendly introduction to the fundamentals of image preprocessing. The concepts are presented in a clear and structured manner.
    CampusX YouTube Playlist
  - **OpenCV Documentation:** This official documentation provides an in-depth overview of various image processing techniques and their implementations in OpenCV. It serves as a valuable reference for practical applications.
    OpenCV Image Processing Tutorials
- **Visualization Aid:** Gain a deeper understanding of how image resizing operates at the pixel level through this insightful video demonstration.
  Image Resizing Explained

Convolutional Neural Netowrks

This section provides useful resources for learning about convolutional neural networks.

Convolutional Neural Networks

This section provides a curated list of resources for learning about Convolutional Neural Networks (CNNs), which form the foundation of many modern computer vision applications.

- **Learning Resources:**
  - **What is a Convolution?** Before diving into CNNs, it's crucial to understand the core idea behind convolution operations.
    Convolution Explained
  - **Introduction to CNNs:** CNNs power most of today's computer vision systems. The following resources cover both foundational and advanced concepts:
    * **NPTEL (by Prof. Mitesh M. Khapra):** A detailed academic playlist explaining CNNs from first principles.
      NPTEL CNN Playlist

    - **Comprehensive Blog:** A beginner-friendly blog that gives an intuitive understanding of CNNs.
      CNN Blog Guide
    - **CampusX (TensorFlow):** A highly accessible and intuitive series, although implemented in TensorFlow. As an exercise, you can rewrite the code in PyTorch.
      CampusX Playlist
    - **CampusX (PyTorch):** A recent addition, this video walks through implementing CNNs from scratch using PyTorch.
      CNN from Scratch in PyTorch
    - **Architecture Exploration:** Try exploring popular CNN architectures like VGG, ResNet, and Inception on your own for deeper understanding.
  - **Advanced NPTEL Course:** For those seeking depth, this advanced NPTEL course is highly recommended. It's perfectly fine if you find it tough initially—perseverance pays off!
    Advanced NPTEL Playlist

- **U-Net Architecture:** U-Net is a critical architecture in semantic segmentation. The following resources cover it in detail:
  - **Quick Overview:** A concise explanation of U-Net in under 10 minutes.
    U-Net in 10 Minutes
  - **Paper Walkthrough:** Detailed walkthrough of the original U-Net paper.
    U-Net Paper Explained
  - **PyTorch Implementation:** A practical implementation of U-Net using PyTorch.
    U-Net PyTorch Code
  - **Playlist:** A complete study of U-Net can be done through this playlist:
    U-Net Playlist
  - **Think about it:** Why was there a need for an architecture like U-Net in the first place.

- **Grad-CAM:** Gradient-weighted Class Activation Mapping (Grad-CAM) is a technique to visualize which parts of an image a CNN focuses on. It belongs to the broader domain of Explainable AI (XAI).
  - **Quick Overview:** A 2-minute overview of Grad-CAM.
    Grad-CAM Summary
  - **Introductory Blog:** A great blog that introduces Grad-CAM along with working code and GitHub demos.
    Grad-CAM Blog
  - **Try out:** Try implementing Grad-CAM yourself—it's a great hands-on learning experience!
  - **Resource:** A detailed guide to Grad-CAM and other Explainable AI techniques.
    Grad-CAM XAI Book

- **Classification:** A basic PyTorch tutorial on image classification. Try building a classifier for the MNIST dataset.
  PyTorch Classification Tutorial
- **Explore:** Classification is just one of many downstream tasks in computer vision. Explore object detection, semantic segmentation, instance segmentation, etc.

Visualization:

- **Types of Convolutions:** To visualize different kinds of convolution operations, check out this video:
  Convolution Types Visualized
- **A Critical View:** You've probably watched a lot of CNN videos by now — here's one that critiques common misconceptions.
  What's Wrong With Most CNN Videos?
- **Beyond the Human Brain:** Even with simple models, we often don't fully understand how they work. This video explains why.
  Deep Models Are Mysterious
- **Visualization Tools:** Here are some cool tools to help you visualize everything you've been learning:
  2D CNN Visualizer, 3D CNN Tool

For those who want to go even deeper, here's an excellent course by Stanford: CS231n — Stanford CNN Course

Assignment

Congratulations on learning the basics of computer vision!! Yes I have said basics since there is a lot for you to learn. The sky is the limit and your first step will be completing this assignment.
This assignment is divided into two tasks -

Task 1

Design and implement a Convolutional Neural Network (CNN) model from scratch. You may choose to develop your own working architecture. Ensure that all architectural decisions are clearly explained and justified — for example, if you choose to use Depthwise Separable Convolutions, provide the rationale behind that choice.

Train the model on the given dataset. If you prefer, you may first train the model on a larger dataset and then fine-tune it on the target dataset. Transfer learning and other training techniques are permitted, but the model must be trained or fine-tuned from scratch.

Task 2

After training, use Class Activation Mapping (CAM) techniques — or any other suitable interpretability method to generate CAM visualizations. These visualizations should help verify whether the model is making decisions based on the correct regions of the input image.
Make sure to clearly state why you chose a specific CAM method and demonstrate a sound understanding of the underlying mathematical principles. This will be assessed in both the LaTeX report and the interview.

Tip: Start with a basic architecture to establish baseline results. Then, iteratively refine and test more complex or efficient models.

Dataset: Link to the dataset to be used - Link

Submission Guidelines

- Create a GitHub repository named `<roll_number>-CSOC-IG` (e.g., `23014019-CSOC-IG`)
- Repository organization:
  - A folder named **GradCam** containing all source code implementations
  - The final report in PDF format, authored using LaTeX

  Everything must be in the github repo itself.
- Submit the repository link via the provided Google Form here
- **Note:** The report constitutes the primary assignment submission. No additional files are required

- **Deadlines are strict and will not be extended**

Final Remarks

Ensure that your submission reflects a clear understanding of the concepts and methodologies applied. Focus on the analytical aspects and the rationale behind your implementations. We look forward to your insightful contributions.

That's it, complete these resources and you will be all ready for CSOC. Hope you will give the resources and assignment your best shot—your future self will thank you!

Adios, *and keep learning!*