BUILD AN AI THAT SEES: Detailed Project Guide for Students

Project Title: Vision AI in 5 Days: From Beginner to Image Recognition Expert.

Project Description:

This project guides participants through building an image recognition model in Python using deep learning. Over five days, students learn fundamental concepts and practical skills in image classification, culminating in a deployable, portfolio-ready toolkit showcasing their expertise in computer vision.

What is This Project About?

The bootcamp is designed to transform beginners into skilled AI practitioners by teaching them how to preprocess images, train convolutional neural networks (CNNs), and build an image recognition system using Python and popular libraries like TensorFlow/Keras. Participants develop hands-on experience with essential machine learning workflows, deepen their understanding of model evaluation, and create professional portfolio assets—preparing them for high-paying roles in data science and computer vision.

Learning Outcomes & Skills Developed:

- Image preprocessing and augmentation
- Deep learning fundamentals and CNN architecture
- Model training, evaluation, and optimization
- Visualization of results and metrics
- Collaboration and documentation best practices
- Showcasing ML projects for recruiters

What You Have to Do (Step-by-Step):

Preprocess and Explore Your Dataset

- Download a free image dataset (e.g., MNIST, Cats vs. Dogs, CIFAR-10 from Kaggle); normalize, resize, and visualize sample images using Python and OpenCV.

Build & Train Your First CNN Model

- Implement a basic convolutional neural network using TensorFlow/Keras in Google Colab, train it on your dataset, and monitor accuracy/loss metrics.

Enhance with Data Augmentation & Model Evaluation

- Apply image augmentation techniques (rotation, flipping) and evaluate your model using metrics like accuracy, precision, recall, F1-score, and confusion matrix. Visualize your results using Matplotlib.

Optimize with Transfer Learning

- Fine-tune a pre-trained model (e.g., MobileNetV2 or similar) on your dataset, compare its performance to your custom CNN, and analyze improvements.

Document, Demo & Submit Your Toolkit

- Prepare a professional GitHub repository: include all scripts, trained models, visualizations, a README, a 30-sec demo video, and a 5-slide evaluation presentation. Participate in the live demo competition.

What to Submit:

GitHub Repository:

- All code scripts (preprocessing, models, evaluation, visualization)
- Trained model files
- Sample test predictions (with expected outputs)
- Visualizations (loss/accuracy curves, confusion matrices, ROC curves)
- Professional README file summarizing your approach and results

Demo Video:

- 30-sec video showing your model performing live predictions

Presentation:

- 5-slide PowerPoint/PDF evaluating model performance/approach

LinkedIn Post Draft:

- A short, recruiter-friendly summary sharing your results

Tips for Students:

- Start simple: Ensure your code runs end-to-end on a basic model before experimenting with advanced architectures.
- Document as you go: Write clear comments, keep a daily log of progress, and focus on making your GitHub repo recruiter-ready.
- Leverage cloud GPUs: Enable GPU in Colab for faster model training (check Runtime > Change runtime type).
- Visualize everything: Always plot training curves and confusion matrices to monitor progress and catch issues early.
- Showcase your skills: Record your demo confidently; clear, professional presentation matters for your portfolio.

Deadline:

Submit your project by **13 Aug 2025, 11:59pm** this submission link - *https://forms.gle/Lpan8GgGmtiLGL4x8*