AI Skills Projects

Objective

Each team will design, train, evaluate, and deploy a **Deep Learning model** using **Convolutional Neural Networks (CNNs)** and **Transfer Learning** to solve a real-world **image classification problem**.

Projects must include experimentation with at least **3 CNN architectures** (e.g., VGG, ResNet, Inception, EfficientNet) and a fully functional **Graphical User Interface (GUI)** that allows users to visualize and test the model.

Team Structure

Each team will consist of **5–6 members** with well-defined roles:

- Data acquisition & preprocessing
- Model building & training
- Evaluation & visualization
- GUI development
- Documentation & report writing
- Repository (GitHub) management

Learning Outcomes

By the end of the project, students will:

- Design and train CNN-based classifiers for real datasets.
- Apply **transfer learning** and **data augmentation** for improved accuracy.
- Compare and analyze multiple network architectures.
- Develop an intuitive **GUI** to showcase results.
- Use **GitHub** collaboratively for version control, documentation, and project management.

General Project Requirements (Grading Breakdown: 20 + 5 Bonus Points)

Component	**Description**	**Points**
Model Development	Dataset preparation, preprocessing, model design, training, and hyperparameter tuning	**5 pts**
Model Evaluation	Accuracy, precision, recall, confusion matrix, comparison between 3 architectures	**3 pts**
Explainability	Grad-CAM or similar visualization to interpret model predictions	**1 pt**
GUI Implementation	A functional GUI supporting image upload or real-time input, showing predictions and confidence	**1 pt**
GitHub Repository	Proper version control with commits from all members, README documentation, clear structure	**2 pts**
Total (Base)	—	**12 pts** (Technical) + **8 pts (Participation/Teamwork)** = **20 pts**
Bonus	Advanced/creative feature (real-time, domain adaptation, mobile deployment, etc.)	**+5 pts**

Bonus Points (5 pts)

Each project includes a **bonus challenge** designed to encourage creativity and technical depth – examples include:

- Real-time inference via webcam.
- Domain adaptation or cross-domain generalization.
- Deployment of a lightweight model (e.g., TensorFlow Lite).

GitHub Requirements (2 pts)

Each team **must use GitHub** for version control and collaboration. To receive full credit:

- Maintain a **well-structured repository** with separate folders for data/, models/, gui/, and docs/.
- All team members must contribute using **individual commits**.
- Include a **comprehensive README** file with:
  - Project title, description, and dataset sources
  - Setup instructions (dependencies, how to run, GUI instructions)
  - Model results and performance graphs
  - Roles of team members
- Maintain a **clean commit history** documenting progress (e.g., data updates, model changes, GUI versions).

Common GUI Requirements

All teams must develop a GUI that provides:

- Image upload or webcam input.
- Display of **top-3 predicted classes** with confidence scores.
- **Grad-CAM visualizations** of the model's attention.
- Comparative results if multiple models are implemented.

- Option to show accuracy or confusion matrix results.
- Designed using frameworks like **Tkinter**, **PyQt**, **Streamlit**, or **Flask**.

🔷 Detailed Project Ideas

---

1. Face Classification

Goal: Identify or verify individuals based on facial images.
Dataset: Labeled Faces in the Wild (LFW), VGGFace2.
Models example:. ResNet50, InceptionV3 (transfer learning)
Details: Implement face detection and alignment; compare models for recognition accuracy.
Applications: Smart attendance systems, identity verification tools.
Bonus: Real-time face recognition using a live webcam feed.

---

2. Gesture Classification

Goal: Classify hand gestures (e.g., “peace”, “stop”, “help”) from images or video streams.
Dataset: Kaggle Gesture Recognition Dataset.
Models example: EfficientNetB0 and ResNet50 for comparison.
Details: Apply strong augmentations to handle lighting and pose variability.
Applications: Contactless control systems, accessibility tools.
Bonus: Implement real-time gesture recognition from webcam input.

---

3. Plant Disease Classification

Goal: Identify plant leaf diseases to aid agricultural diagnostics.
Dataset: PlantVillage Dataset.
Models example: EfficientNetB3, ResNet50, MobileNet.
Details: Use color and orientation augmentations; evaluate performance by disease type.
Applications: Agricultural disease diagnosis systems.
Bonus: Deploy a mobile-ready model using TensorFlow Lite (TFLite).

---

4. Car Type Classification

Goal: Recognize car makes and models from photos.
Dataset: Stanford Cars Dataset.
Models example: InceptionV3 and ResNet50.
Details: Focus on fine-grained features to separate visually similar models.
Applications: Parking management, automotive recognition systems.
Bonus: Add mobile or real-time webcam detection functionality.

---

5. Artwork Classification

Goal: Classify various artwork types (painting, sketch, photography) or artistic styles.
Dataset: WikiArt Dataset.
Models example: VGG16 and EfficientNetB0.
Details: Focus on texture, color, and stroke differentiation.
Applications: Art cataloging, curation, and creative AI tools.
Bonus: Expand from classifying **art style** to classifying **artist identity** (multi-class problem, e.g., Monet, Van Gogh, Picasso).
Use **multi-task learning** to predict both simultaneously.

---

6. Virus Classification

Goal: Classify microscopic virus/bacteria images.
Dataset: Virus Image Dataset (Kaggle).
Models example: EfficientNetB0, ResNet50.
Details: Apply domain adaptation across microscopy conditions.
Applications: Biomedical research support.
Bonus: Evaluate performance improvements after **domain adaptation** or cross-lab generalization.

---

7. Celebrity Face Classification

Goal: Recognize celebrity identities from facial images.
Dataset: LFW, VGGFace2.
Models example: ResNet50
Details: Compare classification accuracy vs. face-embedding matching.
Applications: Media tagging, entertainment automation.
Bonus: Real-time celebrity identification via webcam.

---

8. Satellite Image Classification

Goal: Classify regions in satellite imagery (urban, forest, water, etc.).
Dataset: EuroSAT Dataset.
Models example: InceptionV3 and EfficientNetB0.
Details: Use multi-resolution images and evaluate model generalization.
Applications: Urban planning, environmental monitoring.
Bonus: Implement a **map-view GUI** overlay to visualize prediction outputs.

---

9. Raw Materials Classification

Goal: Classify materials (wood, plastic, metal, glass) based on image texture.
Dataset: Kaggle Materials Image Dataset.
Models example: ResNet18, EfficientNetB1.
Details: Train robust models under lighting/texture variations.
Applications: Industrial sorting, smart recycling systems.
Bonus: Texture Robustness under Varying Lighting. Improve the model's ability to recognize materials even under **different illumination, shadow, or glare conditions**, simulating real factory settings.

---

10. Sign Language Fingerspelling Assistant (ASL Alphabet)

Goal: Recognize American Sign Language (ASL) letters from static images.
Dataset: ASL Alphabet Dataset.
Models: ResNet50, EfficientNetB0, InceptionV3.
Details: Implement augmentations (lighting variation, random backgrounds); include open-set handling for invalid gestures.
Applications: Accessibility technology for the hearing-impaired.
Bonus: Real-Time Robust Sign Detection

🧩 Final Deliverables

Teams must submit:

1. **Code repository (GitHub)** with dataset links, models, and GUI code.

2. **Trained models** and evaluation metrics.

3. **Functional GUI application** (desktop or web-based).

4. **Final report** detailing dataset selection, model comparison, results, and challenges.
5. **Bonus demonstration (if completed)**