Comparative Evaluation of Deep Learning Architectures for Audio and Visual Recognition Tasks

PROJECT PLAN

DEVANSH DEV

CS3822 – BSc Final Year Project

SUPERVISED BY : DR LI ZHANG

DEPARTMENT OF COMPUTER SCIENCE

ROYAL HOLLOWAY, UNIVERSITY OF LONDON

1 Abstract

Audio and video recognition in artificial intelligence are the fundamental building blocks of modern artificial intelligence systems. These two domains form complementary pillars of perceptual intelligence—while audio systems interpret the acoustic environment, visual models analyse spatial and temporal cues in images and video sequences. Speech recognition is extensively utilised in virtual assistants (Siri, Alexa), transcription services, and voice-activated devices, as well as in the healthcare, customer service, and automotive sectors for hands-free engagement. Video recognition is utilised in a plethora of applications, including surveillance, security (facial recognition), autonomous cars, media content analysis, and action recognition in sports or entertainment. In recent years, the same deep-learning architectures used for static image analysis have been extended to video through temporal modelling, enabling systems to recognise motion, activity, and sequential context.

Beyond speech and visual understanding, environmental sound recognition (ESR) has emerged as a significant yet rather uncharted domain within the world of audio-based A.I. It identifies, categorises, and analyses ambient noises; the sounds include ordinary urban noises, natural occurrences such as rain and wind, and mechanical sounds from equipment and cars. ESR involves not only sound identification but also providing context and comprehension of the environment in which these sounds exist. However, research progress was previously limited by the scarcity of suitable and publicly available datasets [1]. The ESC-50 dataset, which included 2000 labelled five-second recordings from fifty classes, established a replicable baseline. The need for more expressive learning techniques was highlighted by the fact that early classical approaches, such as random forests and support vector machines, only achieved about 44% accuracy when compared to 81% human performance [1].

The adoption of deep learning techniques, in particular convolutional neural networks (CNN), materialised due to the limitations of traditional classifiers and revolutionised perceptual modelling in both audio and vision domains [2]. In the visual domain, architectures such as AlexNet and ResNet are also widely employed as backbones for video-recognition models, where temporal information is incorporated through 3D convolutions or recurrent layers to capture motion and dynamic context. This shift paved the way for AlexNet (Krizhevsky et al., 2012), a deep CNN that achieved a top-5 error rate of 15.3% on the ImageNet Large-Scale Visual Recognition challenge and demonstrated the power of hierarchical feature extraction by dramatically outperforming all prior methods. AlexNet's success inspired many ESR studies to apply transfer learning from pretrained CNNs to spectrogram "images", which gave major performance gains over traditional features.

A model trained on a large source dataset can be reused for a smaller task through transfer learning. With this technique, a network can retain the low-level features it learnt (edges, textures, frequency patterns) while only retraining its higher-level layers to adapt to the new data. Computational resources and the amount of training data required are significantly reduced with this approach while often improving generalisation, especially when the target dataset is limited in size. In convolutional networks, the early layers capture features that tend to be more generic and transferable across domains, and the deeper layers become increasingly specialised for the original task [3]. Due to this concept, architectures such as AlexNet [2] and ResNet-50 [4] have become the standard for transfer learning experiments across both audio and video. Similar architectures are also used in modern video-recognition systems, where spatial feature extractors are extended with modules to model

motion [4]. By evaluating both audio and video modalities, this project will examine how transfer-learning architecture generalises across perceptual domains and where modality-specific actions are required.

For audio data, waveforms of sound are transformed into two-dimensional spectrograms, representing the distribution of frequencies over time. This enables convolutional networks to process sound in a manner comparable to image recognition. The research that followed introduced enhanced training strategies such as Between-Class (BC) learning, in which two sounds from different classes are mixed to improve separability and feature geometry [5]. This shows the rapid progress of sound-recognition models but exposes a need for systematic, cross-modal evaluation.

The aim of this project is to fill that gap by empirically comparing three models: a baseline CNN constructed from scratch, a transfer learning model based on AlexNet [2], and a deeper ResNet-50 [4] architecture. For term 1, all experiments will be implemented in Python using PyTorch and Librosa, with the ESC-50 dataset serving as the benchmark. Performance of the models will be evaluated using accuracy, area-under-curve (AUC), and confusion matrix analysis to assess overall and class-specific recognition performance [6]. The outcomes of my project are expected to show architectural depth, residual design, and transfer learning recognition accuracy and generalisation of features across modalities. The findings of this study will facilitate further extending this framework from environmental sound to video-based recognition in term 2, inspiring the broader understanding of multimodal deep-learning systems and their potential application in our daily lives.

2 Timeline

My project will be executed across two phases: Term 1,I will focus on implementing and evaluating audio-based models, then during Term 2, I will extend the framework to video and multimodal recognition. Barring any major setbacks, I plan to finish early and explore additional experiments on cross modal learning or attention mechanisms.

2.1 Term 1

● **Week 1**: Project familiarisation and literature review. Review work on ESR, transfer learning and CNNs (AlexNet, ResNet). Study the ESC-50 dataset and preprocessing methods.

● **Week 2**: Learn Librosa, spectrogram generation, and data augmentation techniques (noise, pitch shift, and time stretch).
● **Weeks 3–4**: Build an end-to-end baseline (CNN + preprocessing) to validate the workflow and catch integration issues early.
● **Weeks 5–6**: Implement AlexNet, then extend to ResNet-50. Add model checkpointing, early stopping and standardised evaluation (accuracy, AUC, confusion matrix).
● **Week 7**: Train models with multiple random seeds for statistical validity. Document performance trends and generalisation gaps, and enable TensorBoard/Weights & Biases for training visualisation and monitoring.

● **Weeks 8–9**: Run structured experiments with tracked hyperparameters (learning rate, batch size, augmentation). Integrate Between-Class (BC) learning if feasible. Conduct error analysis on misclassified samples to understand failure modes.

● **Weeks 10-11**: Analyse architectural depth impact and residual connections. Buffer for debugging and reruns; finalise interim report and presentation. Ensure all code, logs, and results are reproducible.

3 Risks and Mitigations

As with any project, especially a machine learning one, risks are inherent. They could falter progress and impact outcomes in a negative manner. This section will outline the possible challenges my project may have to face and the strategies I have devised to mitigate them.

3.1 Hardware Failure and Data Loss

Local hardware failure could mean the loss of trained models, experimental and documentational logs and code progress, significantly hampering the project. My solution for this will be to maintain all code with regular commits to the department GitLab repository. All trained model checkpoints, experiment results, configuration files and logs will also be stored in the GitLab repository.

3.2 Computational Resource Constraints

Training deep networks such as ResNet-50 may exceed available computational resources, leading to long training times or memory limitations. I will secure GPU access early (Colab, university HPC or personal hardware). If the resource constraint problem persists, I shall implement mixed precision training, reduce batch size or use gradient accumulation.

3.3 Poor Time Estimation and Task Overruns

Understanding task complexity could cause schedule problems, particularly in weeks 3-6 (implementation) and weeks 8-9 (experimentation). To avoid this, I will follow weekly deliverables and log progress in Gitlab diary.md. Hold regular supervisor check-ins for guidance. I will prioritise core deliverables such as comparing the three models over optional explorations.

3.4 Model Convergence and Performance Issues

Networks may overfit, fail to converge or achieve poor performance due to architectural flaws or improper hyperparameters. To mitigate this, I will implement early stopping, learning rate scheduling and cross validation. I will monitor training curves intensively using TensorBoard. If a model does underperform, I will conduct a systematic debugging: verify data quality and review architecture choices against literature.

3.5 Scope Creep and Feature Overload

Attempting too many exploratory ideas (BC learning, attention mechanisms, cross-modal extensions) could make me lose focus and compromise core deliverables. Core requirements such as implementation of the baseline CNN, AlexNet and ResNet-50 are the highest priority. Optional enhancements to the project will only be pursued once if core work is on track. I will not chase marginal improvements at the expense of detailed evaluation and report quality.

3.6 Overhead of Audio and Video Knowledge

As this project is at the cross section of computer science and audio and video signal processing, gaps in understanding spectrograms, audio augmentation, acoustic properties or video temporal models could lead to substandard choices or misunderstanding of the results. The first two weeks are dedicated to learning Librosa and audio preprocessing fundamentals. For term 2, I will similarly review literature about video preprocessing and temporal modelling techniques. If I am unclear about any domain-specific decisions, I will consult my supervisor.

3.7 Unbalanced Report and Code Development

Spending the majority of my time on implementation could leave insufficient time for quality report writing or vice versa, resulting in incomplete findings and documentation. To solve this problem, I will write the literature review and interim report progressively across the whole term rather than just deferring all the writing to the last few weeks.

3.8 Reproducibility and Code Quality Issues

Untracked changes of code or missing seed control could make results non-reproducible, complicate debugging and reflect negatively on the project's professionalism. I will follow the best practices, such as modular code, clear function documentation and a comprehensive README with proper setup instructions. Configuration files will store hyperparameters separately from the code. I will periodically test reproducibility by re-running experiments from scratch in a clean environment.

Acronyms

AI	Artificial Intelligence
ESR	Environmental Sound Recognition
CNN	Convolutional Neural Network
ML	Machine Learning
AUC	Area Under Curve
BC	Between-Class (Learning)
GPU	Graphics Processing Unit
HPC	High-Performance Computing
SE	Software Engineering
VESTA	Visualization for Electronic and Structural Analysis *(a 3D visualization program for structural models and volumetric data)*
ResNet-50	Residual Network (50-layer variant, architecture using residual connections)
GitLab	Web-based Git repository manager *(also a name, not an acronym)*

References

[1] Piczak, K.J. (2015) ‘ESC: Dataset for Environmental Sound Classification’, in *Proceedings of the 23rd ACM international conference on Multimedia*. New York, NY, USA: ACM, pp. 1015–1018. Available at: https://doi.org/10.1145/2733373.2806390.

[2] Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2017) ‘ImageNet classification with deep convolutional neural networks’, *Communications of the ACM*, pp. 84–90. Available at: https://doi.org/10.1145/3065386.

[3] Yosinski, J. *et al.* (2014) ‘How transferable are features in deep neural networks?’ Available at: https://doi.org/10.48550/arxiv.1411.1792.

[4] Kaiming He *et al.* (2016) ‘Deep Residual Learning for Image Recognition’, in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, pp. 770–778. Available at: https://doi.org/10.1109/CVPR.2016.90.

[5] Tokozume, Y., Ushiku, Y. and Harada, T. (2017) ‘Learning from Between-class Examples for Deep Sound Recognition’. Available at: https://doi.org/10.48550/arxiv.1711.10282.

[6] Fawcett, T. (2006) ‘An introduction to ROC analysis’, *Pattern recognition letters*, 27(8), pp. 861–874. Available at: https://doi.org/10.1016/j.patrec.2005.10.010.

[7] Zhang, L. *et al.* (2022) ‘Sound classification using evolving ensemble models and Particle Swarm Optimization’, *Applied soft computing*, 116. Available at: https://doi.org/10.1016/j.asoc.2021.108322.