# Roadmap

## 1. Clear Task Definition

☑ ~~Choose which commands to classify: for example, the standard 10 keywords (`yes`, `no`, `up`, `down`, etc.), possibly including `unknown` and `silence`.~~

- We use the full dataset even if it is more complex and heavy.

☑ ~~Check class balance and consider data balancing techniques if necessary.~~

---

## 2. Initial Modeling (Baseline)

☑ ~~Implement a **basic CNN model**, inspired by [Sainath15]:~~

- Input: 40×101 log-Mel spectrogram (already prepared)
- Architecture: 2-3 convolutional layers + fully connected layers
- Output: multi-class classification

This will serve as your **reference baseline** to compare more advanced models.

---

## 3. Experimentation with Advanced Architectures

Once the baseline is working, experiment with:

### CNN-Based

- Residual CNN (ResNet block)
- Inception-style convolutions (as suggested in the course guidelines)

### RNN-Based

- CNN + BiLSTM to capture temporal dynamics (preferred)
- CNN + GRU

### Transformer-Based

- Implement a **Speech-Transformer** based on the paper
  - Initial convolution layers + positional encoding + attention-based encoder-decoder

    - Optional: 2D-Attention across time and frequency dimensions
    - Find something to differentiate it from the one of the paper !

### GAN-like Hybrid

- CNN or Transformer as feature extractor + a GAN-style discriminator as classifier (more experimental but I think it could be interesting!)

## 4. Optimization for Edge Devices

Once I have 5–6 models evaluated for pure performance (accuracy, loss), explore model compression techniques:

- **Pruning**: TensorFlow Model Optimization Toolkit
- **Quantization** (post-training or quant-aware training)
- **Knowledge Distillation**: distill a larger model into a smaller one
- **Batching**, caching and efficient dataset handling

### Compare:

- Average inference time
- Memory usage
- Final model size in MB

## 5. Evaluation and Visualization

- Metrics: Accuracy, Precision, Recall, F1 Score
- Confusion matrix
- T-SNE (look for it) or PCA visualization of learned feature representations to show class clustering
- Spectrogram + prediction visualizations (qualitative error analysis)

## 6. Report and Presentation

- Write the report in LaTeX (use the Moodle template)
- Include:

  - Motivation for each architecture
  - Design choices
  - Explanatory figures and model comparisons
  - Final analysis of complexity and memory usage

## Bonus Ideas (optional but valuable)

- Use an autoencoder for feature extraction (use bottleneck vectors for classification)
- Train on raw audio using WaveNet-style convolutions
- Add attention layers on top of CNN or LSTM, see:

Spoken_SQuAD.pdf