BRAINHACK TIL 2021: TEAM DELTA

Team members:

- Nguyen Cao Duy
- Tran Xuan An
- Tan Song Ze
- Lance Nathan Beltran Mosura
- Bryan Chua Bing Huan

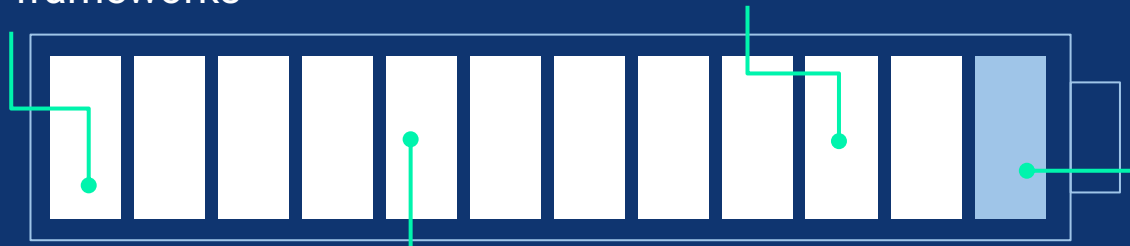
PRESENTATION OUTLINE

INTRODUCTION

Challenge recap &
Overview of frameworks

AUDIO CLASSIFICATION

Challenge 2, 4, 6



OBJECT DETECTION

Challenge 1, 3, 5

CONCLUSION & QA

CHALLENGE RECAP

OBJECT DETECTION

- Create bounding boxes around the animals in the photo
- Classify the detected animals into different classes such as ‘Snake’, ‘Dog’, etc.

AUDIO CLASSIFICATION

- Classify the sound files into difference single word commands such as ‘One’, ‘Forward’, ‘Snake’, etc.

FRAMEWORK OVERVIEW

FASTAI

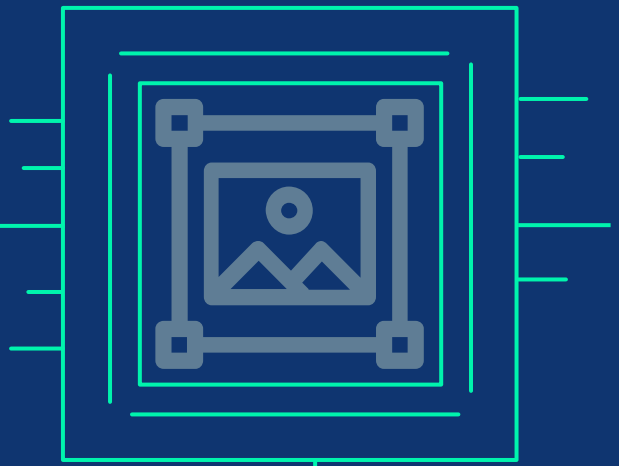
- A deep learning library built on top of PyTorch which provides high-level components that can quickly and easily provide state-of-the-art results.

IceVision

- An agnostic fastai-like framework specific for computer vision

FASTAUDIO

- An extension of fastai for audio-related tasks



OBJECT DETECTION

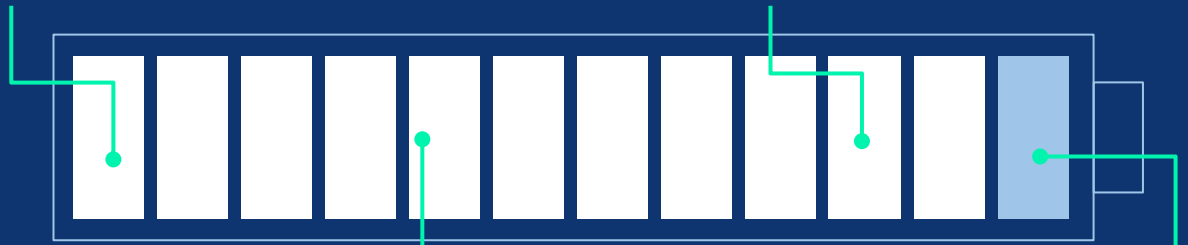
PIPELINE

DATA PREPARATION

External data

MODEL TRAINING

YOLOv5



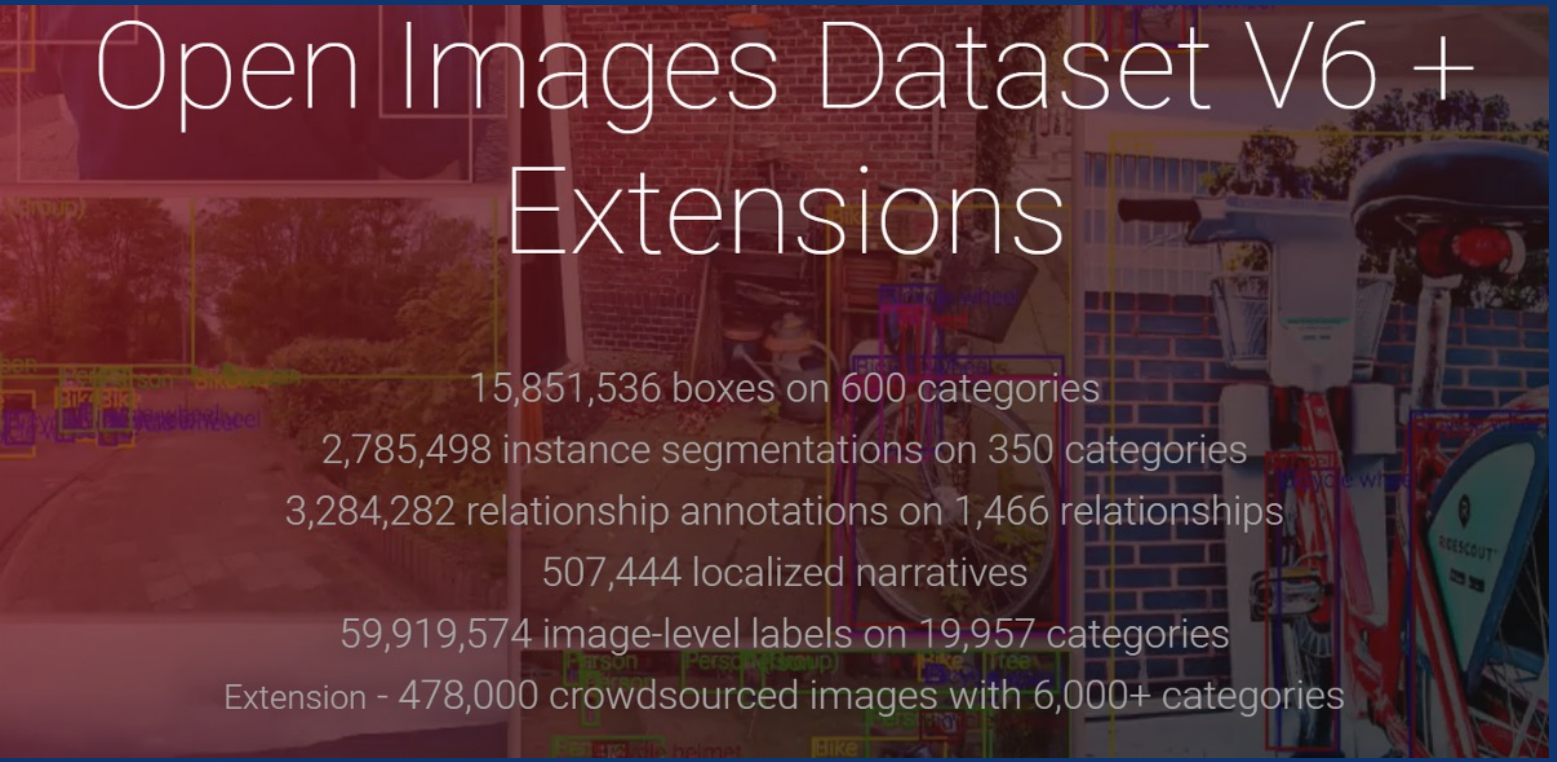
DATA PREPROCESSING

Resize (384) & Data augmentation

PREDICTION

Restore bounding boxes

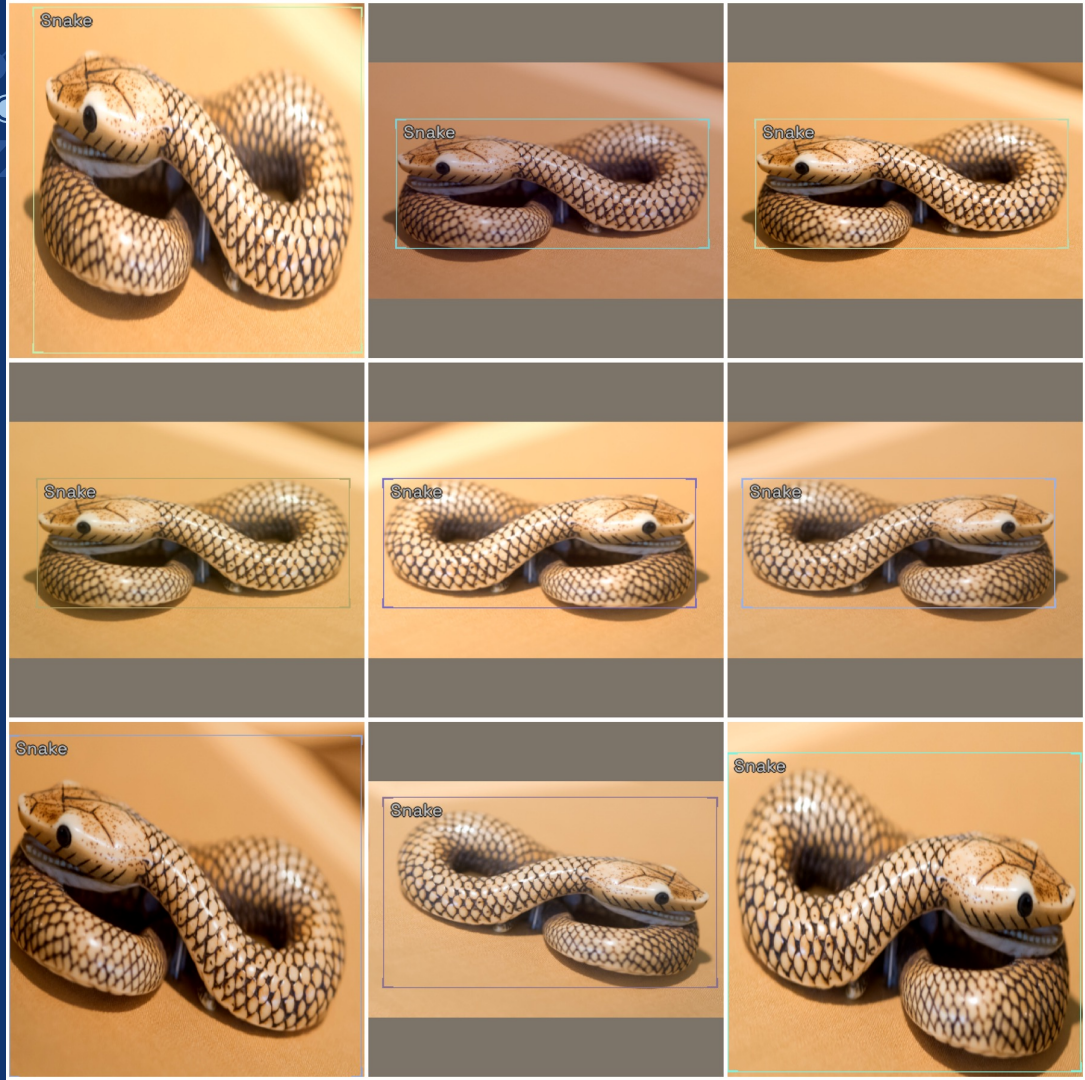
EXTERNAL DATA



- Increase the original dataset by 10-20 times!

DATA AUGMENTATION

- HorizontalFlip
- ShiftScaleRotate
- RGBShift
- RandomBrightnessContrast
- Blur



MODEL TRAINING

- Different pretrained models were considered: RetinaNet, Faster R-CNN, EfficientDet
- Chosen model: YOLOv5 (extra-large)
- Best training/testing time & mAP
- Pretrained on the COCO dataset (from `ultralytics`)

MODEL TRAINING

- Optimizer: Adam
- Loss function: BCEWithLogitsLoss (class probability and object score)
- Focal Loss was also considered to address the imbalanced dataset

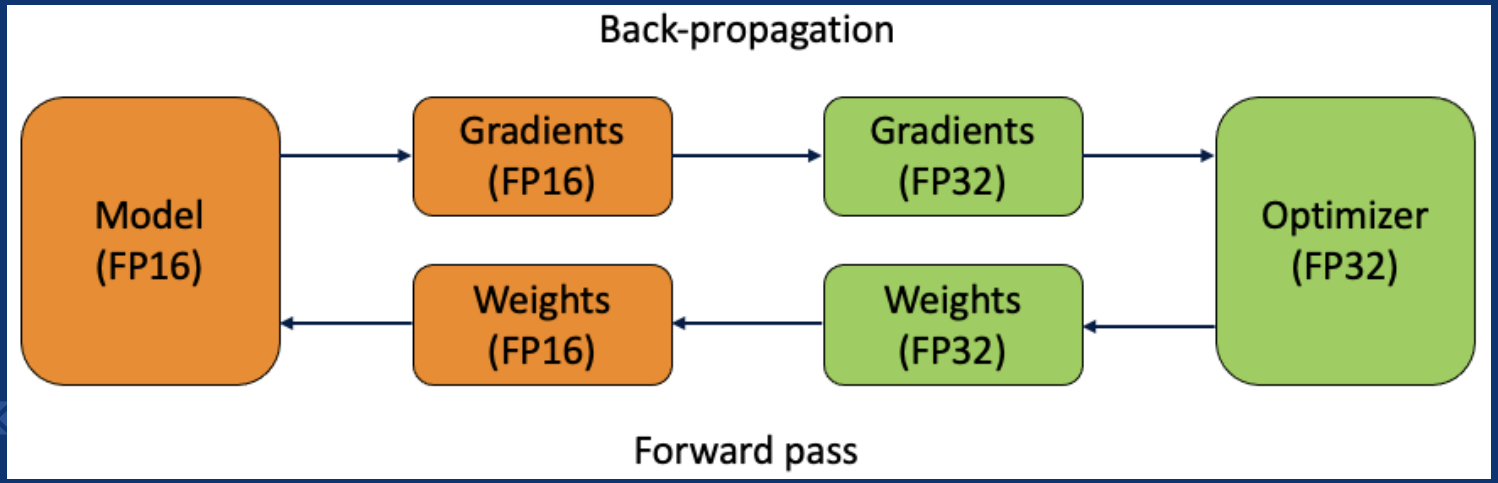
- 2-stage training:
- 1) Freeze the model body, train the model head for a few epochs
- 2) Unfreeze the model body, then train the entire model for at least 30 epochs

- Mixed precision training was used to reduce memory usage (explained later)
- Learning rate finder was used to find the most optimal learning rate (explained later)
- 1cycle training policy was also used (explained later)
- Save model callback is used to log the model with the best validation loss

MIXED PRECISION TRAINING

- Main idea: forward pass and gradient computation in half precision fp16 (to go fast), but backpropagation in single precision fp32 (to maintain precision)

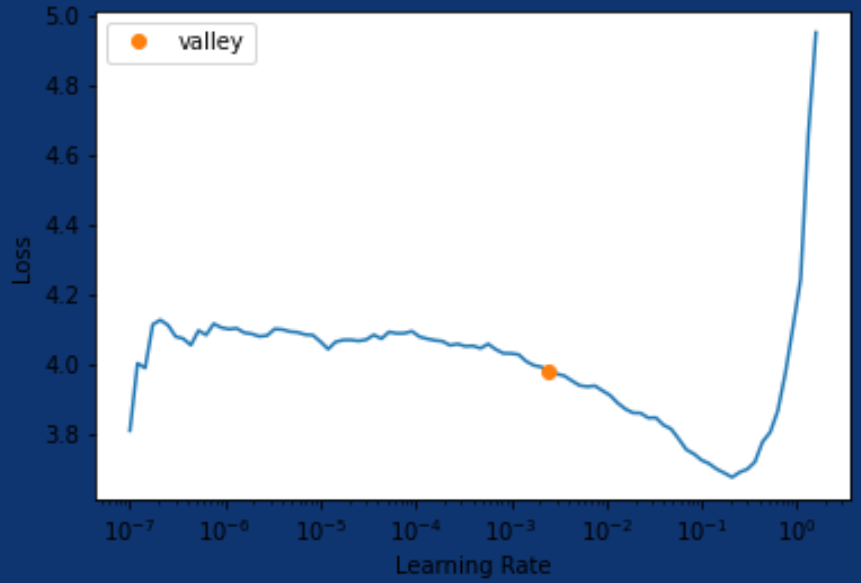
- Benefits:
- 1) Only 1 line of code change with `fastai` training loop
- 2) Allows us to increase our batch size, thus reducing training time



LEARNING RATE FINDER

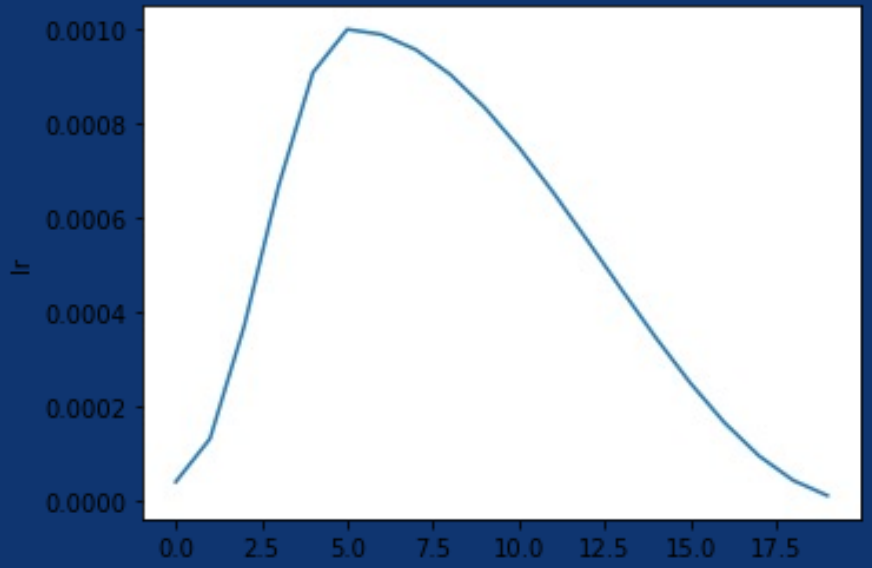
- Main idea: launch a mock training on a few batches of date while varying the learning rate and plot the correspondence training loss

- Benefit: systematically determine the optimal learning rate without guessing



1CYCLE TRAINING POLICY

- Main idea: increasing learning rate at the beginning, then decreasing learning rate
- Benefit: faster convergence



AUDIO CLASSIFICATION

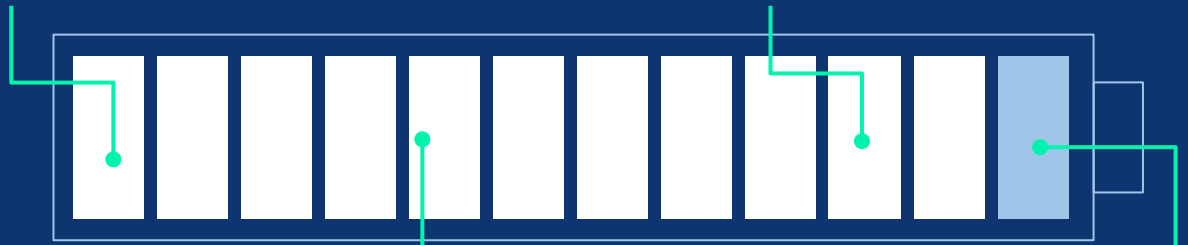
PIPELINE

DATA PREPARATION

External data

MODEL TRAINING

ResNet, DenseNet, etc.



DATA PREPROCESSING

Mel spectrogram (1000ms)
& Data augmentation

PREDICTION

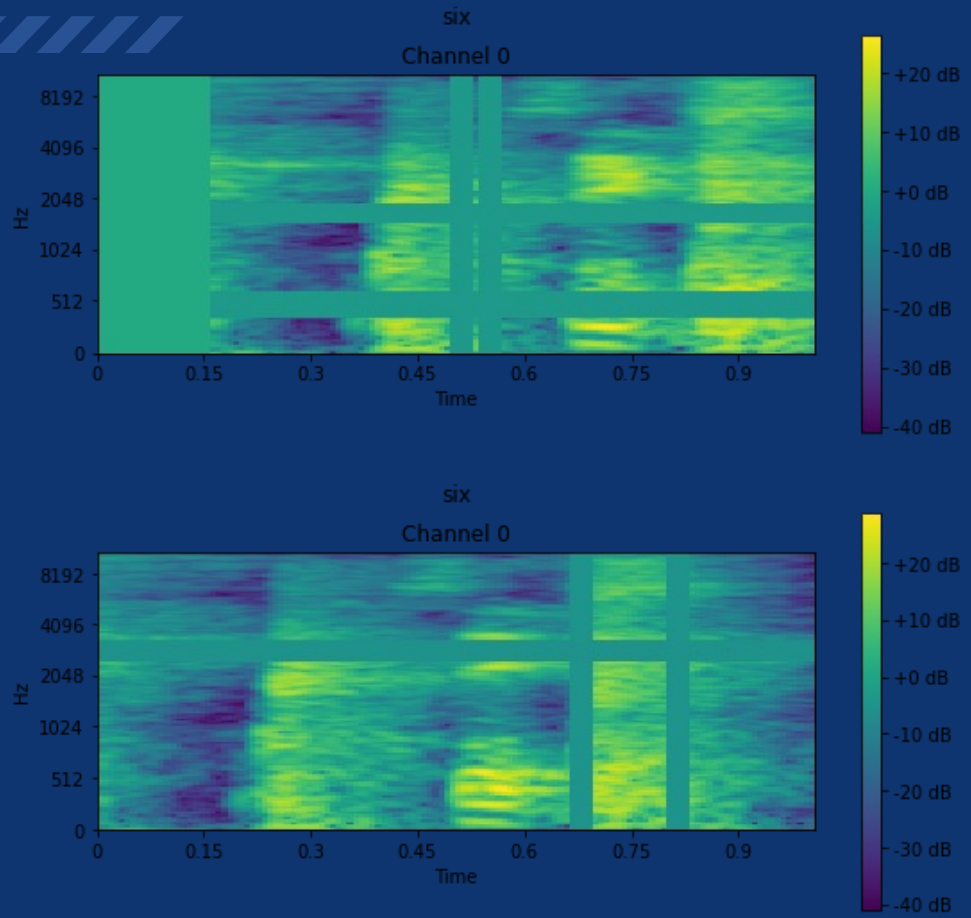
Test time augmentaion &
Ensemble

EXTERNAL DATA

- Two additional datasets: **Free Spoken Digit Dataset (FSDD), Speech Commands**
- Increase the original dataset by 5-10 times!

DATA AUGMENTATION

- MaskTime
- MaskFreq
- SignalShifter
- ChangeVolume
- SignalCutout



MODEL TRAINING

- 5 different models were chosen

- 3 were pretrained on ImageNet: ResNet, DenseNet, VGG
- 2 were trained from scratch: XResNeXt, XSE_ResNet
- => To have different diverse models for ensembling

- Optimizer: Ranger (explained later)
- Loss function: CrossEntropyLoss

- 1-stage training:
- Unfreeze the model body, then train the entire model for at least 15 epochs

- Similar task-independent technique were also used: mixed precision training, learning rate finder, 1cycle training policy, save model callback

RANGER OPTIMIZER

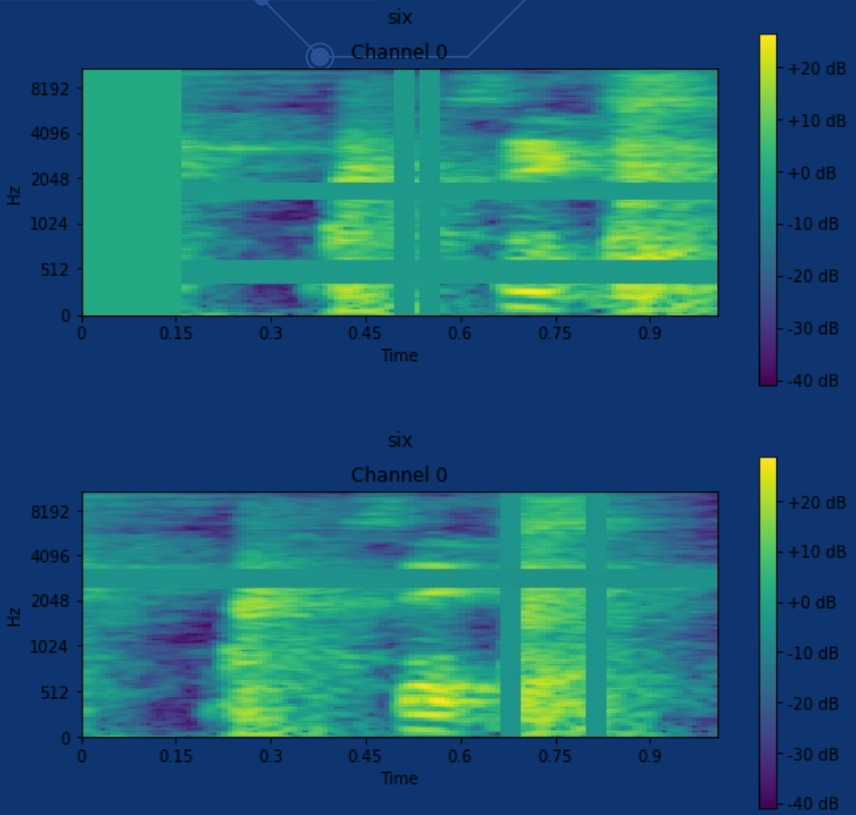
- Main idea: a combination of Rectified Adam (RAdam) with Lookahead optimizer

- RAdam is an improved version of Adam regarding the adaptive momentum mechanism
- *“On the Variance of the Adaptive Learning Rate and Beyond”*

- Lookahead is a safety mechanism to reduce variance in training
- *“Lookahead Optimizer: k steps forward, 1step back”*

- Benefit: better convergence compared to Adam (especially when training models from scratch)

TEST TIME AUGMENTATION (TTA)



- Random augmentation transforms are applied on each image before prediction
- Repeat this process for n times
- Calculate the average prediction probabilities

MODEL ENSEMBLING

- Main idea: average the predictions of all 5 models after TTA (n = 5)
- The final prediction for each image is a combination of **25** different predictions!

Model	Max Accuracy (Validation) (c1)
resnet18	0.98687
densenet121	0.98856
vgg19	0.98899
xresnext18	0.98645
xse_resnet18	0.98772
ensembling	0.99168

THANK YOU FOR ATTENTION

Q & A

We would love to answer your questions!