Master in
Computer Vision
Barcelona

Final Presentation

Module: C5: Visual Recognition

Group: 7

Students: Cristian Gutiérrez
Iñaki Lacunza
Marco Cordón
Merlès Subirà

Index

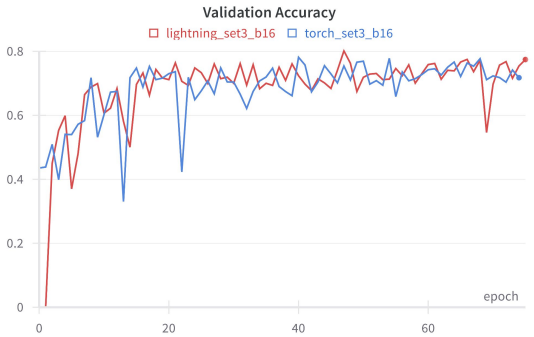
1. **Summary of previous weeks.**
2. Week 6: Multimodal Human Analysis
   - Task a. Different sets statistics
   - Task b. Define a classifier with image only and evaluate
   - Task c. Define a train strategy
   - Task d. Define a test strategy
   - Task e. Define a strategy to represent acoustic data
   - Task f. Define a strategy to represent text data
   - Task g. Multimodal model
   - Task h. Ablation study
   - Task i. Conclusions

Week 1: Comparing modules

PyTorch vs PyTorch Lightning

Results

Lightning achieved slightly better results than PyTorch

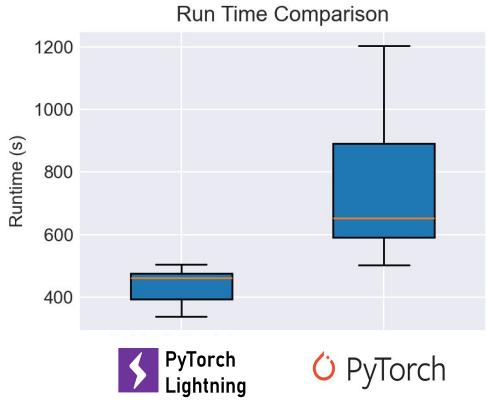


	PyTorch Lightning	PyTorch
Best Val	80.16%	77.84%

Both achieve similar results

Performance

12 Runs for each framework



	PyTorch Lightning	PyTorch
Avg	7.23 mins	12.49 mins

Lightning is **faster**
Lightning is more **stable**

Code

PyTorch lets you code

```
optimizer = torch.optim.Adam(model.paramete
for batch in dataloader:
    optimizer.zero_grad()
    inputs, labels = batch
    outputs = model(inputs)
    loss = loss_function(outputs, labels)
```

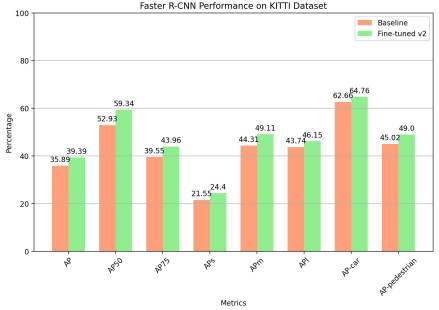
Lightning has a convention

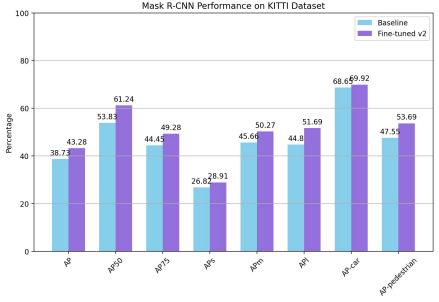
```
class LitModel(pl.LightningModule):
  def training_step(self, batch, batch_idx):
    inputs, labels = batch
    outputs = self(inputs)
    loss = self.loss_function(outputs, labels)
    return loss
```

PyTorch is more **flexible**
Lightning is more **opinionated**

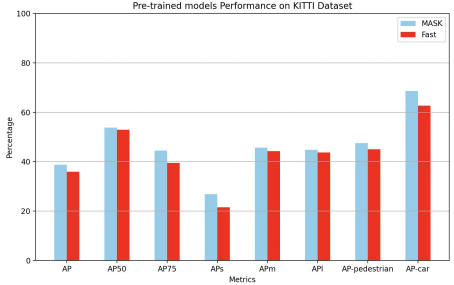
Week 2: Object detection and segmentation

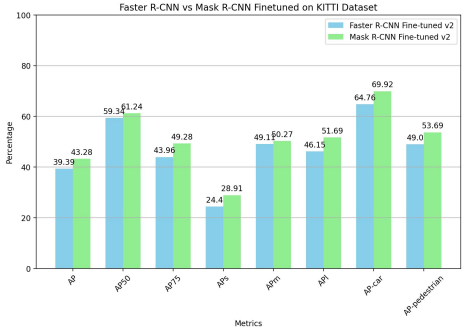
Pre-trained vs fine-tuned





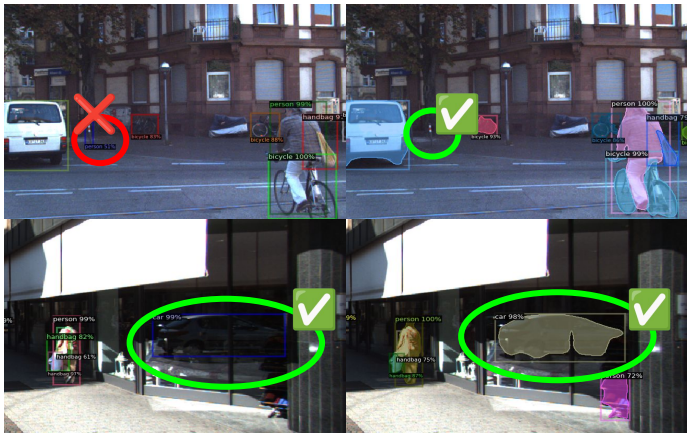
Faster vs Mask





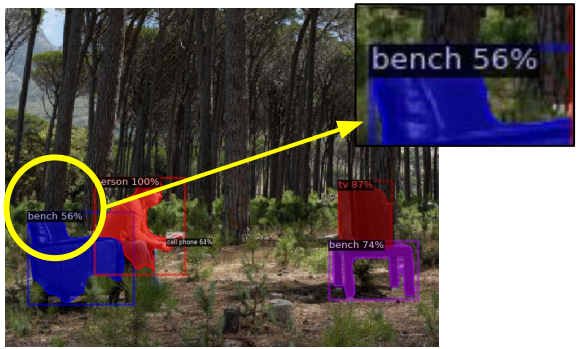
Qualitatively

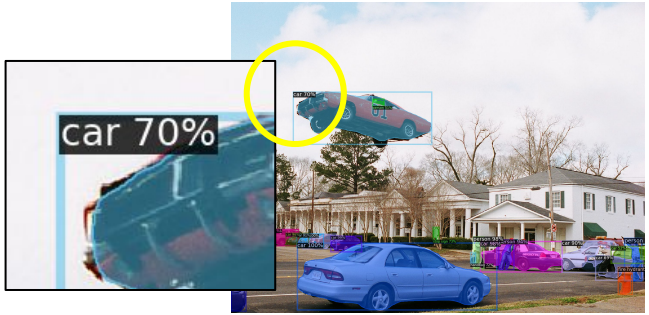
Faster R-CNN Mask R-CNN



Week 2: Out of context object recognition

Case 1: Well classified but low score





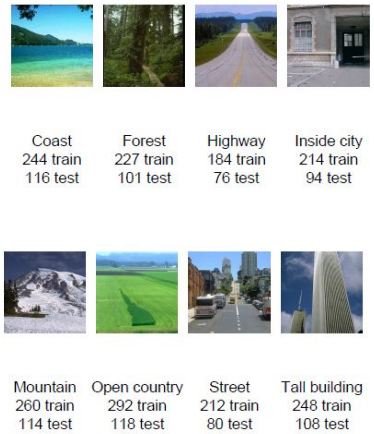
Case 2: Wrongly classified

Case 3: not detected

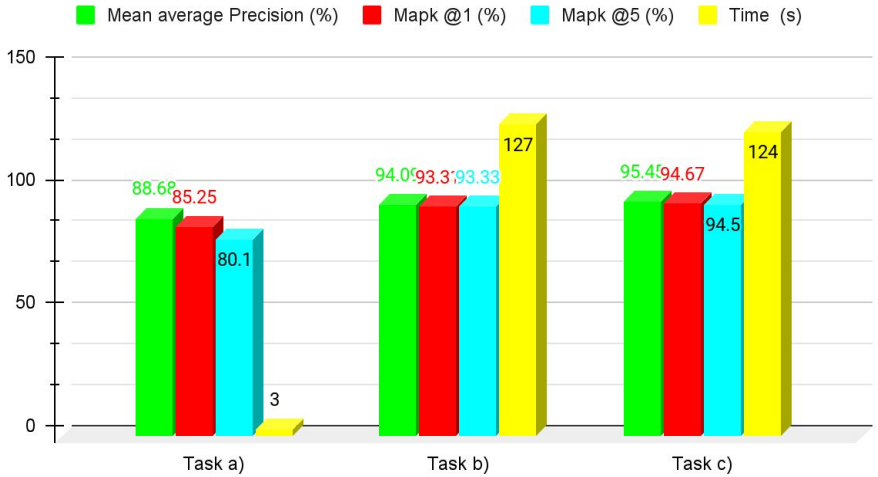
Week 3: Image retrieval with metric learning

MIT_SPLIT dataset
ResNet-18

8 classes



Score and time comparison for:
a) ResNet-18. b) Siamese network. c) Triplet network.



Query image: street

Retrieved images and distance (euclidean):

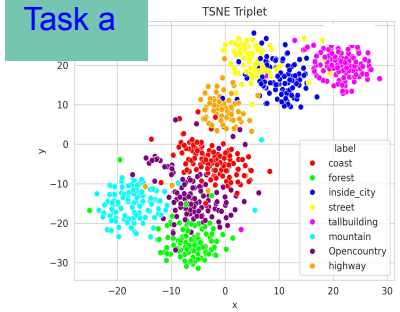
a) 11.55
b) 5.87
c) 5.87

a) 11.99
b) 6.73
c) 3.45

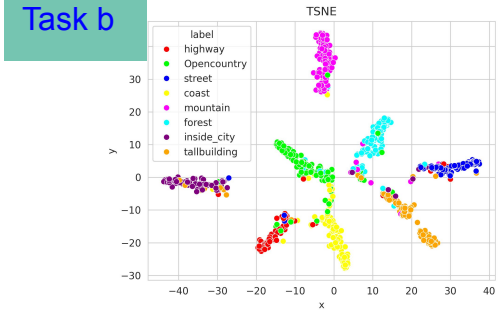
Week 3: Image retrieval with metric learning

TSNE

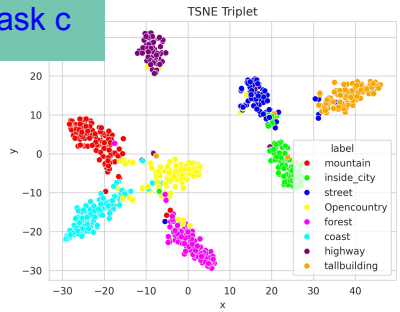
Task a



Task b

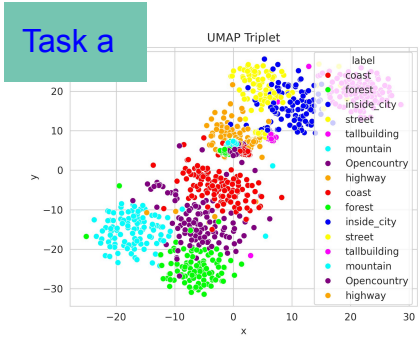


Task c

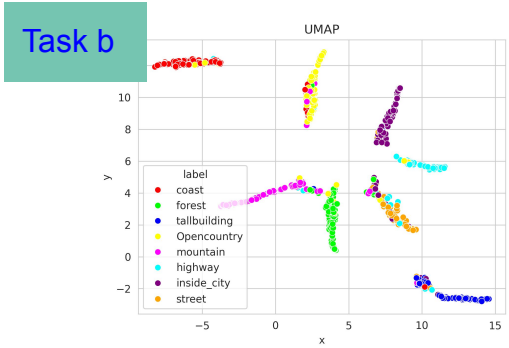


Umap

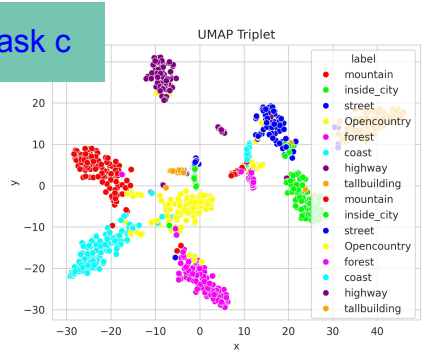
Task a



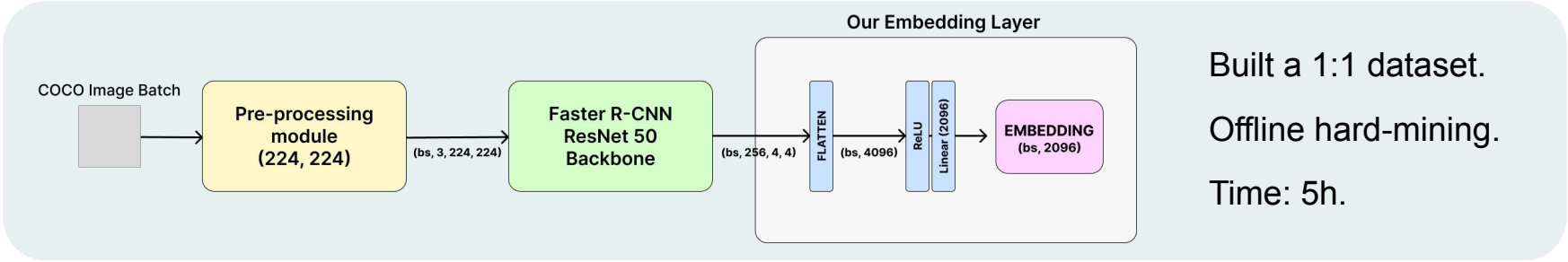
Task b



Task c



Week 3: Image retrieval with metric learning

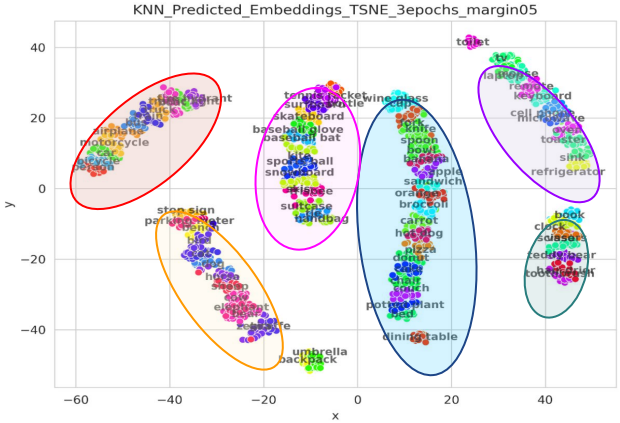


Built a 1:1 dataset.

Offline hard-mining.

Time: 5h.

Image Retrieval on COCO with Faster R-CNN. Val. accuracy: 97.65%. Total run time: 6h 30min.



Found super classes:

LIVING AND TRANSPORT
(person, truck, motorcycle…)

SPORT ACTIVITIES AND MISC
(skateboard, baseball bat, skis…)

COMMON UTILITIES
(i.e. book, scissors, toothbrush…)

FOOD AND KITCHEN
(i.e. broccoli, knife, spoon…)

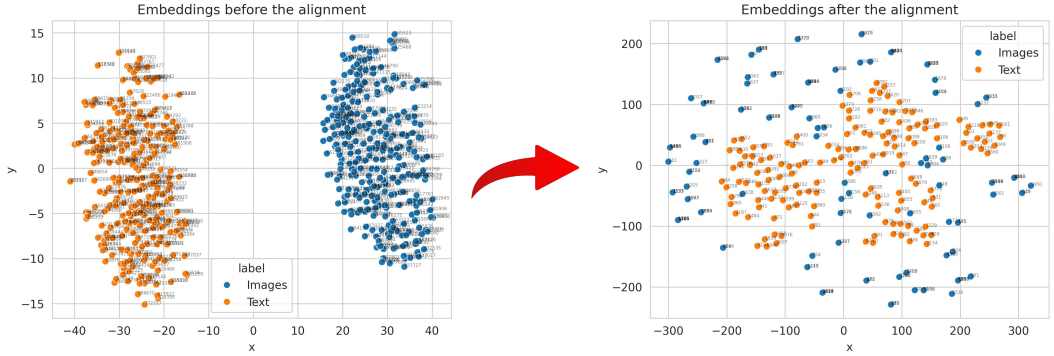
ELECTRONICS
(i.e. keyboard, cell phone…)

ANIMALS AND URBAN
(i.e. elephant, dog, stop sign…)

Week 4: Cross-Modal Retrieval: Image to text

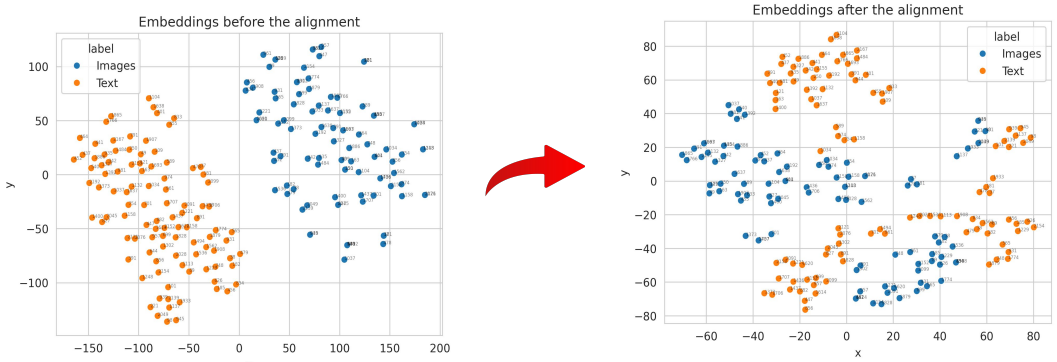
Image to text

- Train time: 14:57h, using a RTX 3090.
- Final loss value: 0.079.



Text to image

- Train time: 13:32h, using a RTX 3090.
- Final loss value: 0.075.



(100 instances picked randomly)

Week 4: Cross-Modal Retrieval: Image to text

Image to text

Image with ID 564133

Ground Truth Caption

Street signs on a pole above traffic lights.

Predicted Captions

A graffiti-ed stop sign across the street from a red car

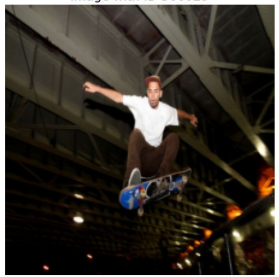
A vandalized stop sign and a red beetle on the road

A red stop sign with a Bush bumper sticker under the word stop.

The parachutes fly through the sky next to each other.

A bike on the platform of a public transportation stop at night.

Image with ID 506029



Ground Truth Caption

A man in the air on his skateboard doing a trick.

Predicted Captions

A girl trusts another girl to shave her while sitting back to back.

A boy holding his arm around the shoulders of the girl sitting next to him.

A boy places his arm around a friend.

A female student is talking on her phone while walking.

A woman is shaving her face while sitting on a wooden bench.

Text to image

Old picture of woman cooking together in the kitchen.

GT Image

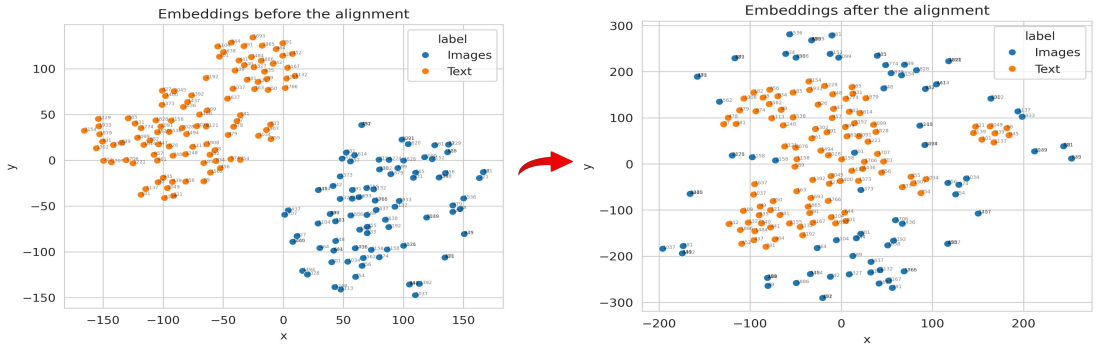
two ladies holding a large bowl filled with doughnuts

GT Image

Week 4: Cross-Modal Retrieval: Using BERT

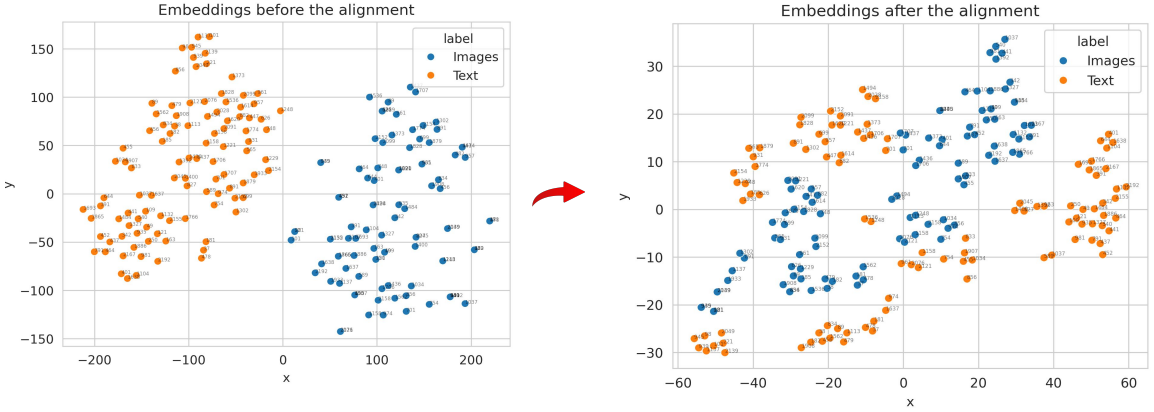
Image to text

- Train time: 6:45h.
- Final loss value: 0.089 (vs 0.079).



Text to image

- Train time: 6:14h.
- Final loss value: 0.091 (vs 0.075).



Week 4: Cross-Modal Retrieval: Using BERT

Image with ID 640307

Ground Truth Caption

A bed with a pile of clothing on top of it.

Predicted Captions

Three cats sleeping with their owners on a bed.

Thee cats sleeping on a bed with a person.

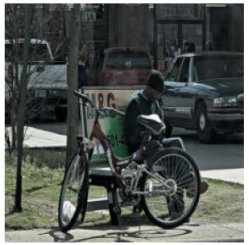
a white bear laying on a bed with someone's hand on the other side

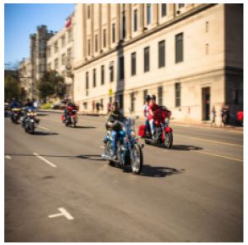
A stuffed animal is laying on the bed by a window.

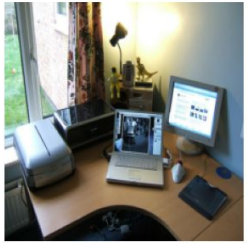
A cat sitting on a bath rug next to a pile of clothes.

A group of people on public transportation stare at their phones.

GT Image







Week 5: Diffusion models

SD 2.1

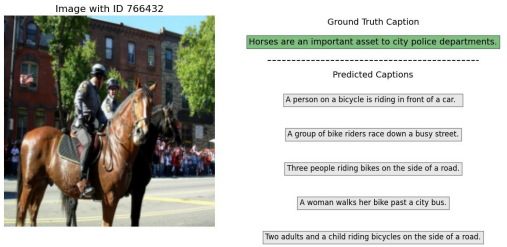
SD XL

SD 2.1 Turbo

SD XL Turbo

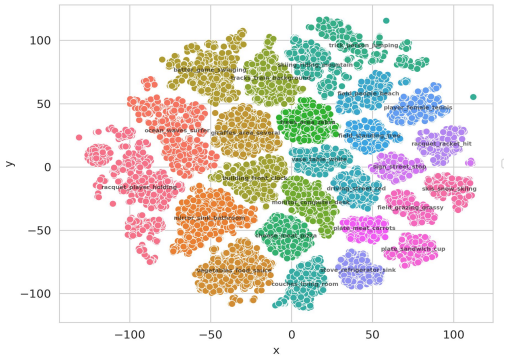
SD XL w/ 500 inference steps and DDIM

Problem: animals

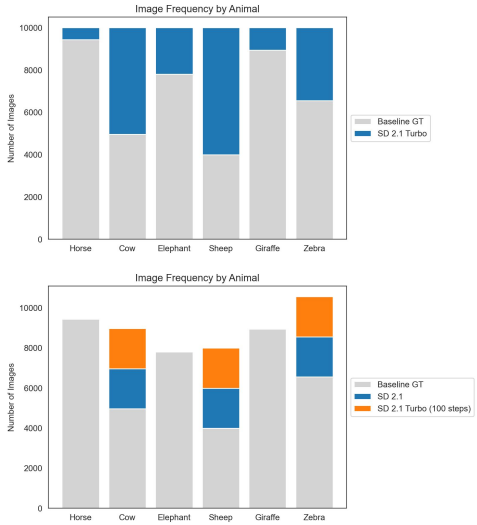


Sentence Transformer ✓

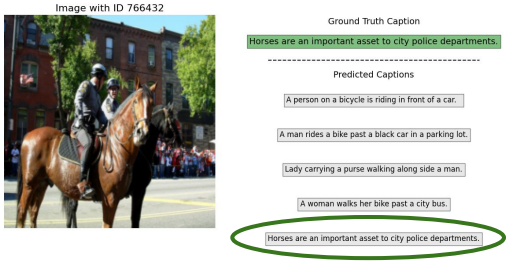
BERTopic ✗



2 Different Data-Augmentation Approaches



With data augmentation and fine-tuning



Index

1. Summary of previous weeks.
2. **Week 6: Multimodal Human Analysis**
   - Task a. Different sets statistics
   - Task b. Define a classifier with image only and evaluate
   - Task c. Define a train strategy
   - Task d. Define a test strategy
   - Task e. Define a strategy to represent acoustic data
   - Task f. Define a strategy to represent text data
   - Task g. Multimodal model
   - Task h. Ablation study
   - Task i. Conclusions

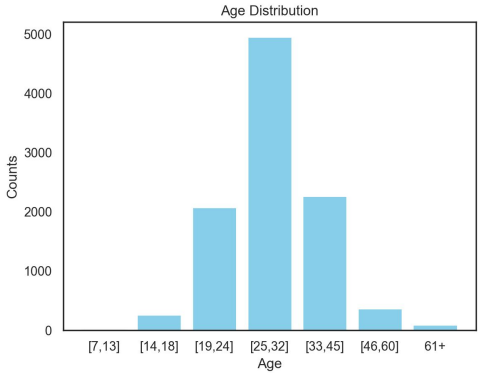
Task a. Different sets statistics

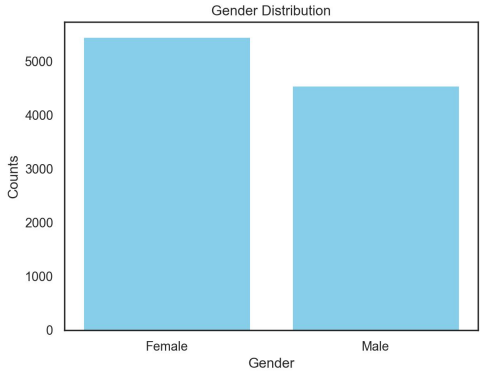
Task a. Different sets statistics

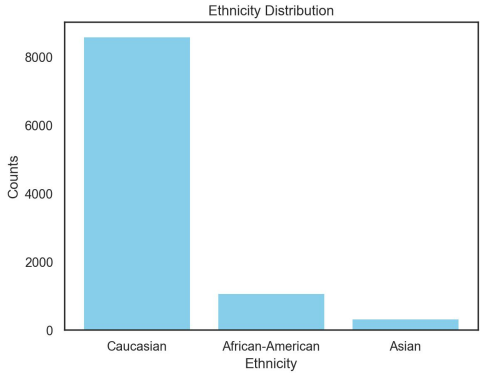
First thing we did was to create our own `DataLoader` class and take into account the full dataset without partitions to compute the Histograms for each label AgeGroup, Gender and Ethnicity.

```
full_dataset = ConcatDataset([train_partition, val_partition, test_partition])
```

Then we first computed the frequencies of each individual objective variable:



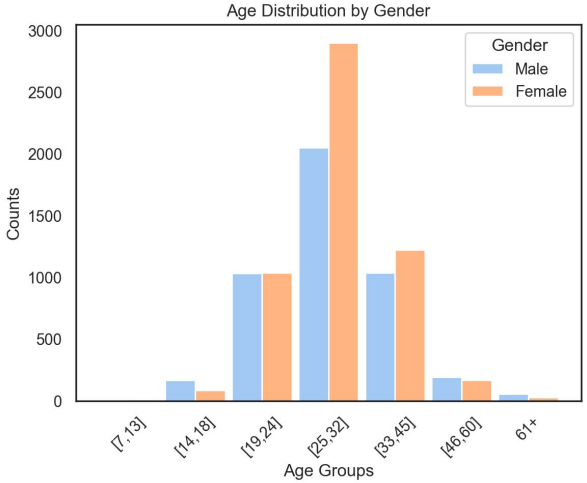




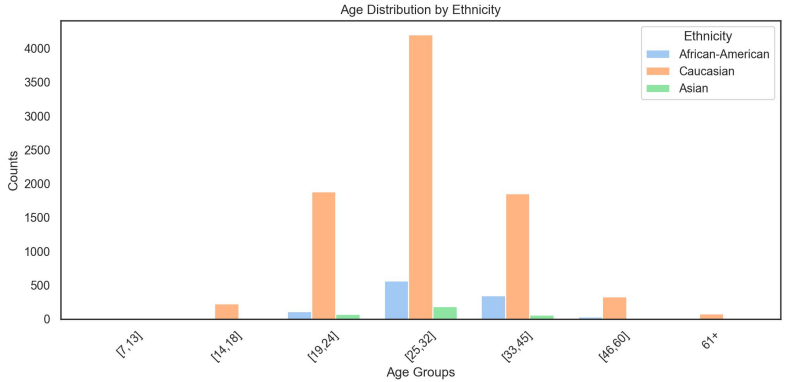
The age distribution seems to follow a normal distribution, by having the majority of the data centered around the central age range [25, 32]. Also, there is a huge predominance on Caucasian in the Ethnicity. On both `train`, `val` and `test` partitions this claims we've just made for the fully hold.

Task a. Different sets statistics

We then computed the **age distribution** by our other two objective variables ‘Gender’ and ‘Ethnicity’.

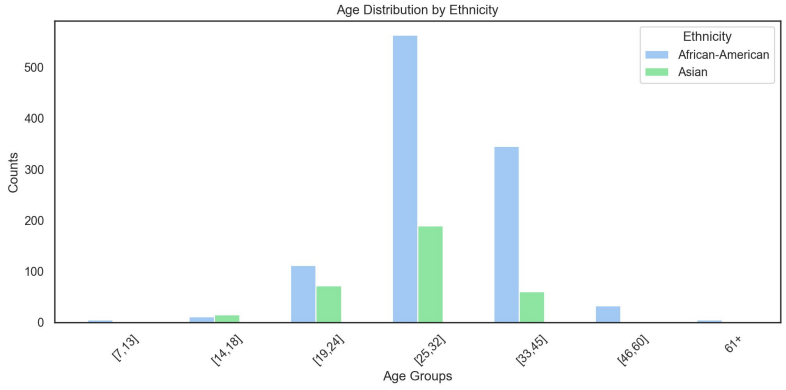


Both genders follow a similar **normal**-like distribution. Also, they are more or less balanced, thus we do not draw any problems.



Both genders follow a similar **normal**-like distribution. Also, they are more or less balanced, thus we do not draw any problems.

Age Distribution without Caucasian



For clarity, we plot the age distribution for the Ethnicity case without Caucasian. We can see the **imbalance** in age for African-American subjects.

Task a. Different sets statistics

We indicate in underscore maximum of each age range, and we add the difference of each age with respect to the baseline (the ethnicity that has the maximum frequency) per each range.

We also mark in yellow those cases where there is NO representation.

Ethn.	7–13	14–18	19–24	25–32	33–45	46–60	61+
Caucasian	14 (+0%)	228 (+0%)	1885 (+0%)	4201 (+0%)	1854 (+0%)	328 (+0%)	80 (+0%)
African-A.	5 (-64%)	11 (-95%)	112 (-94%)	563 (-87%)	345 (-81%)	33 (-90%)	5 (-94%)
Asian	0 (-100%)	15 (-93%)	72 (-96%)	189 (-96%)	60 (-97%)	0 (-100%)	0 (-100%)

We can see how we don't have **any representation** for **Asian** subjects of ages **[7–13], [46–60] and 61+**.

Furthermore, by the observed differences we can see dangerous levels for the same age ranges of African-Americans, and for both ethnicities on [14-18] years.

Task a. Different sets statistics

Gender / Ethn.		7–13	14–18	19–24	25–32	33–45	46–60	61+
F	Caucasian	8 (+0%)	72 (+0%)	919 (+0%)	2378 (+0%)	929 (+0%)	149 (+0%)	29 (+0%)
	African-A.	4 (-50%)	7 (-90%)	69 (-92%)	390 (-84%)	244 (-74%)	19 (-87%)	0 (-100%)
	Asian	0 (-100%)	6 (-92%)	49 (-95%)	134 (-94%)	50 (-95%)	0 (-100%)	0 (-100%)
M	Caucasian	6 (+0%)	156 (+0%)	966 (+0%)	1823 (+0%)	925 (+0%)	179 (+0%)	51 (+0%)
	African-A.	1 (-83%)	4 (-97%)	43 (-96%)	173 (-91%)	101 (-89%)	14 (-92%)	5 (-90%)
	Asian	0 (-100%)	9 (-94%)	23 (-98%)	55 (-97%)	10 (-99%)	0 (-100%)	0 (-100%)

We can see how **African-A** is under-represented, and within them **Male** is worse represented than **Female**.

Asian is the worst ethnicity in terms of representation, and **Male** is worse represented than **Female** also.

Task b. Train a classifier with image only and evaluate

Task b. Image only classifier

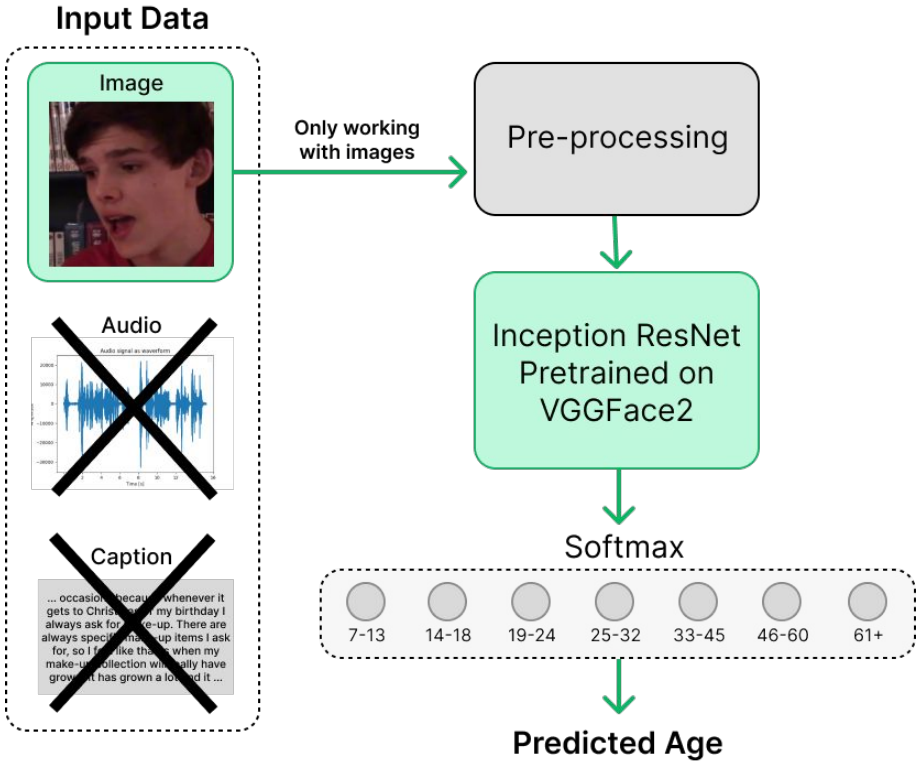
MODEL OVERVIEW

We have used the `baseline Inception Resnet V1` model given to us to do this task. The model uses pre-trained weights for the large-scale face recognition VGGFace2 dataset containing +3.3 million samples.

As pre-processing, all images are resized to a common size of 224, then several data augmentations `TrivialAugmentWide` and a normalization step for each of the images.

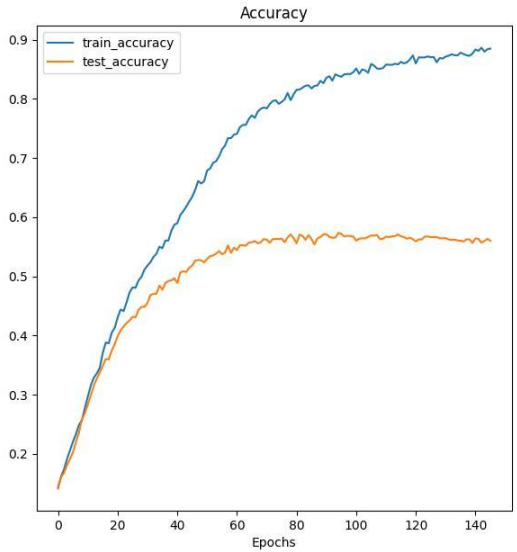
For training, the images are batched with a size of 256, and 300 epochs with an early stopping of 50.

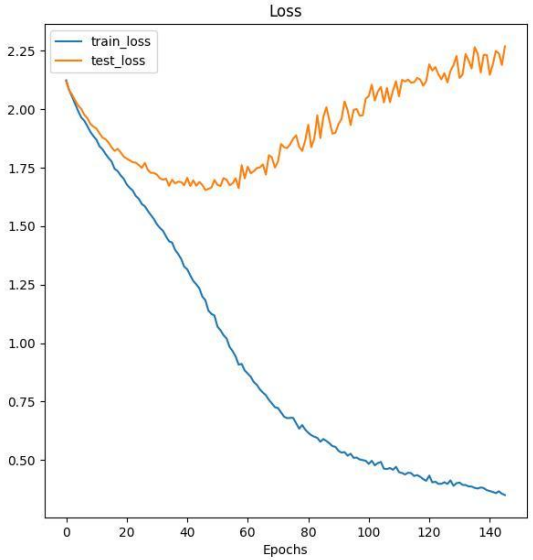
For the loss, Cross-Entropy func is used to compute the logits, which are then processed through a softmax function to yield the final age predictions.



Task b. Image only classifier

If we monitor the loss obtained for both stages of *training* and *testing* we can see how for the loss, the model is able to converge for the training partition, however for the test set, the loss fluctuates (which is somewhat normal) but overall it increases as the training loss decreases. This is a first indicative sign of **overfitting**!



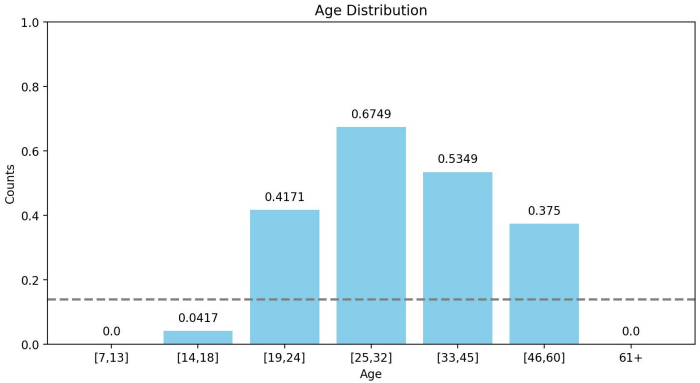


Overall, both the loss and the accuracies look inefficient. There is overfitting being suffered possibly memorizing some training instances that are not being translated onto testing knowledge.

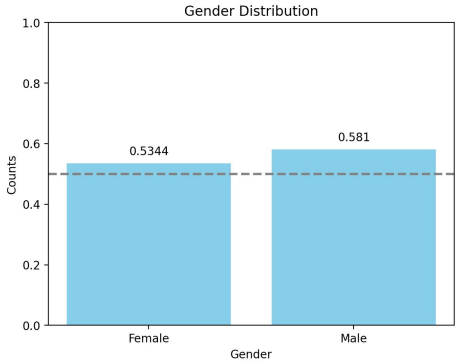
Task b. Image only classifier

In terms of evaluation, we used the `evaluate.py` script to assess the performance of the model. This script calculates the global average accuracy as well as category-specific accuracies based on age, gender, and ethnicity. *(We indicate with a dash line better than chance)*

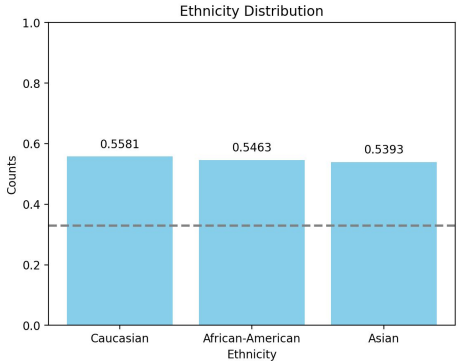
AGE CLASSIFICATION ACCURACIES



ACC BY GENDER



ACC BY ETHNICITIES



Also, we obtain a mean age bias of **0.12** which might indicate some discrepancy on the fairness of the classifications results, more precisely we obtain 0.3305 of bias for Caucasian, which is explained by the vast difference in frequencies of samples w.r.t. other ethnicities.

Task b. Image only classifier

Having a batch size of 256 is quite big, and it is generally recommended to always use the biggest batch size our GPU accepts as a rule of thumb. But we wanted to check how much this hyperparameters affects and we re-trained the model just by changing the batch size to 32.

What if we use a smaller batch size of 32?

DEFAULT BATCH SIZE OF 256

- Epoch until stop: **145**
- Avg Test Accuracy: **0.5559**

TRAINING TIME: 1h 13min

✅

BATCH SIZE OF 32

- Epoch until stop: **77**
- Avg Test Accuracy: **0.2906**

TRAINING TIME: 37min

❌

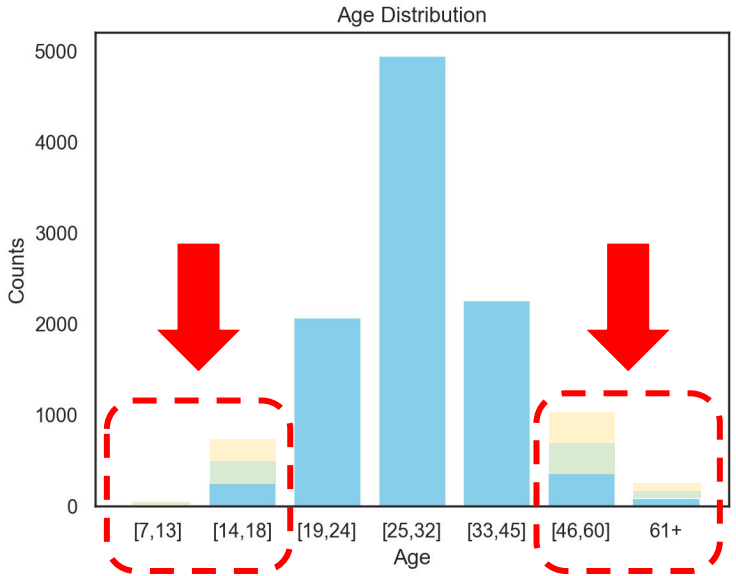
Thus, even though our early stop criteria triggers much faster in advance with a low batch_size, the model has not been able to generalize at all, at the cost of having a big leap in accuracy.

Task c. Define a train strategy

Task c. Define a train strategy

Analysing the task_a results, we have confirmed the big unbalance that our dataset has.

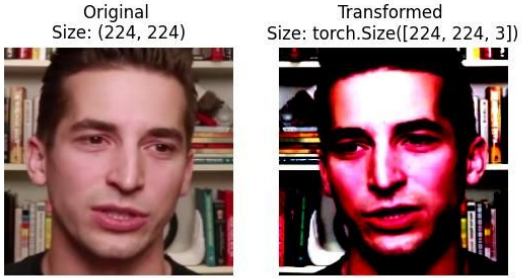
Data Augmentation



Old train set: **6011**

New train set: + 457+ 457 = **6925**

First new test set: (457 images)

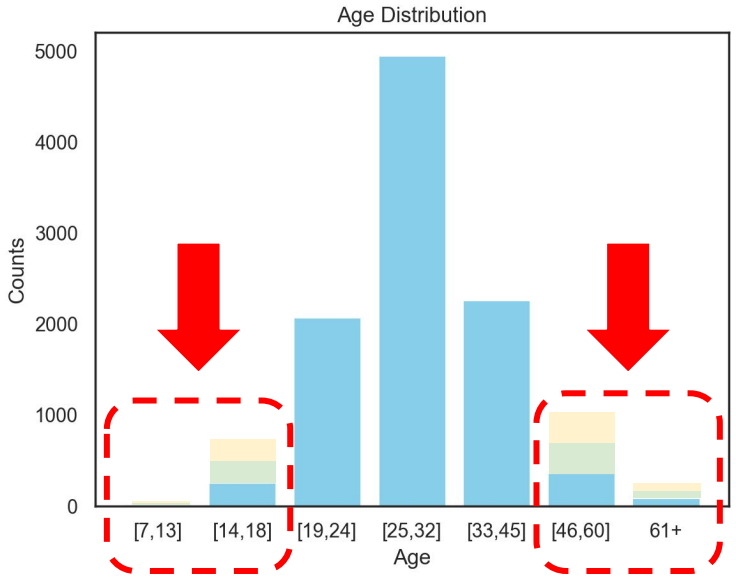


Second new test set: (457 images)

Task c. Define a train strategy

Analysing the task_a results, we have confirmed the big unbalance that our dataset has.

Data Augmentation



Old train set: **6011**

New train set: + 457+ 457 = **6925**

Custom Loss

Weight vector inversely proportional to the number of train images of that class.

1 ➗ ([30, 492, 1264, 2932, 1353, 696, 153] ➗ 6925)

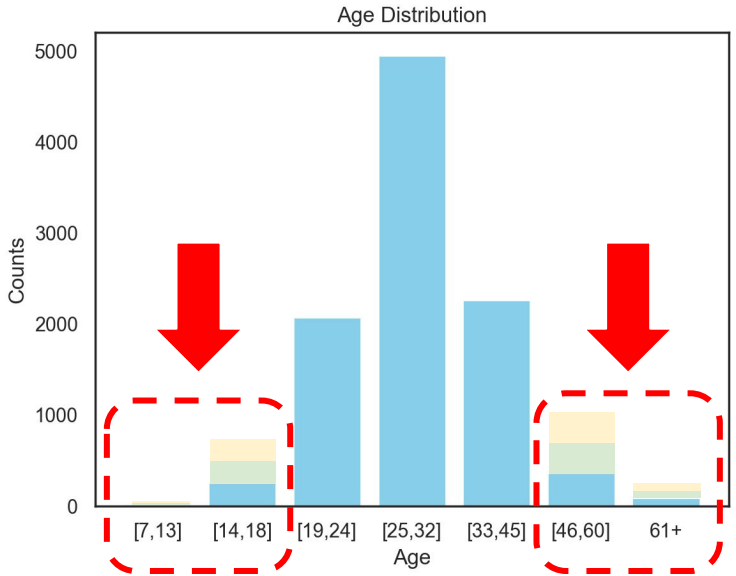
↓

[230, 14, 5, 2, 5, 10, 45]

Task c. Define a train strategy

Analysing the task_a results, we have confirmed the big unbalance that our dataset has.

Data Augmentation



Old train set: **6011**

New train set: + 457+ 457 = **6925**

Custom Loss

Weight vector inversely proportional to the number of train images of that class.

1 ➗ ([30, 492, 1264, 2932, 1353, 696, 153] ➗ 6925)

↓

[230, 14, 5, 2, 5, 10, 45]

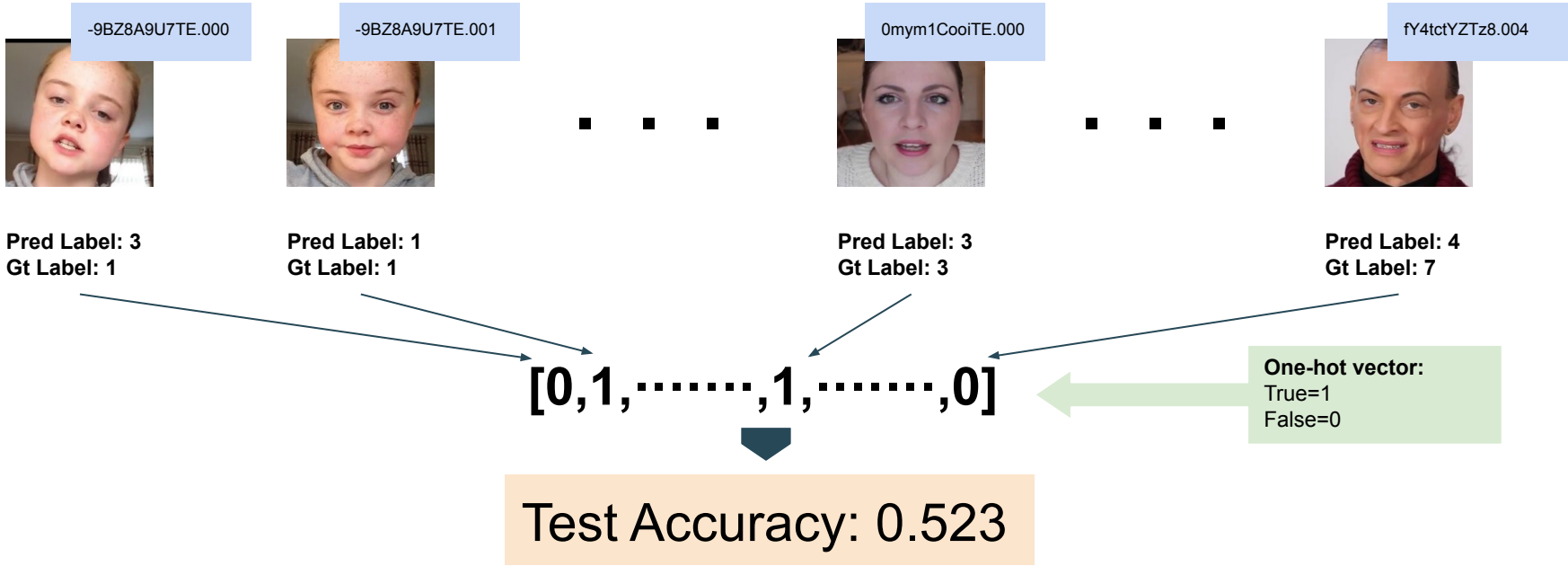
Baseline Train Strategy (Test Acc):	**0.52**
New Train Strategy (Test Acc):	**0.54**

Task d. Define a test strategy

Task d. Define a test strategy

Baseline strategy: **treat all the images of each user individually** as if they were not related to each other.

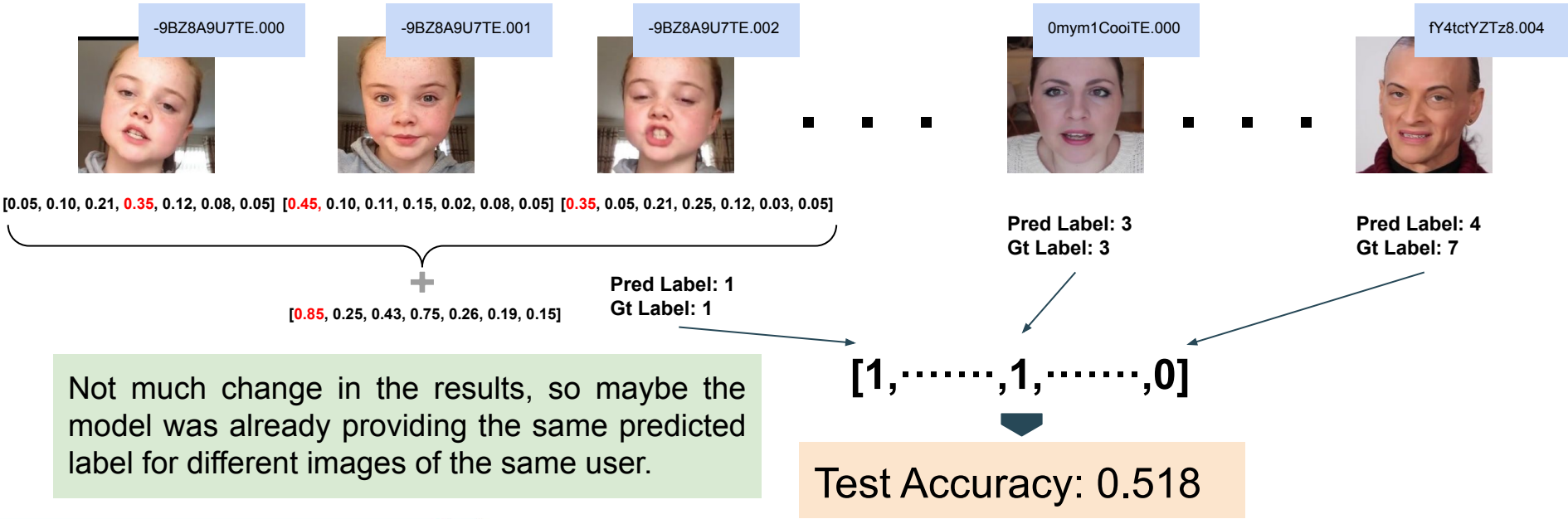
At the end, aggregate all the test images predicted results to calculate the **Test Accuracy**:



Task d. Define a test strategy

Aggregate results of the user to avoid false predictions.

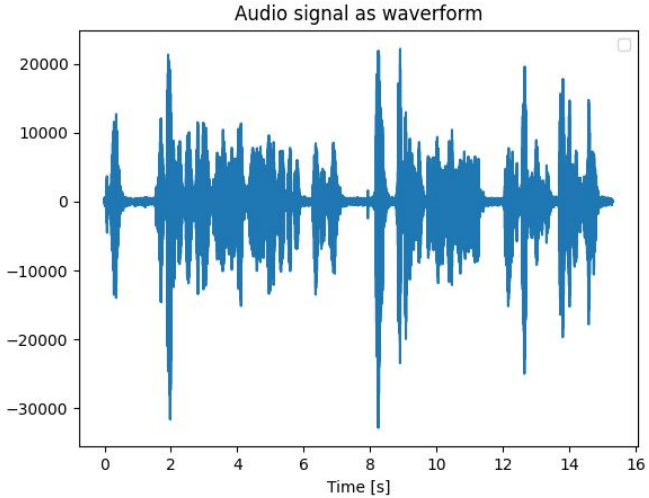
We added the probabilities of the same user and kept the highest one.



Not much change in the results, so maybe the model was already providing the same predicted label for different images of the same user.

Task e. Define a strategy to represent acoustic data

Task e. Define a strategy to represent acoustic data

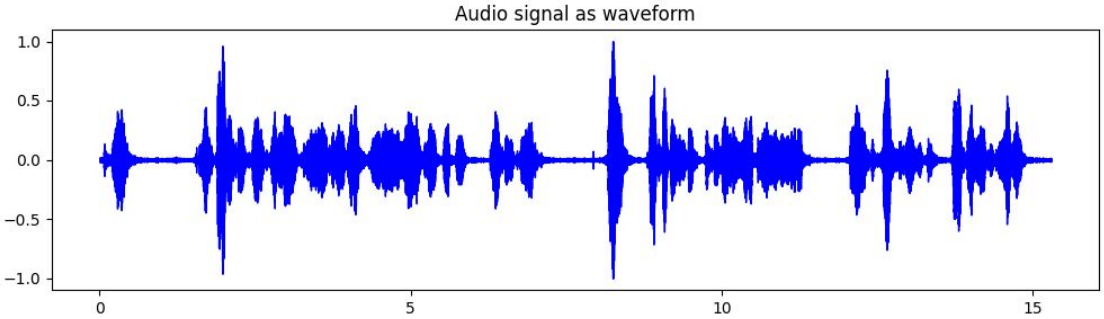


Used Librosa to extract audio information.

'Normalization' by dividing the original values by 2^15-1 (because we have 16bit audios):
Better way of working with the data, since the saved volume may be different depending on the used device.

Plot computed using '***wavfile***'

Plot computed using ***'Librosa'***



Task e. Define a strategy to represent acoustic data

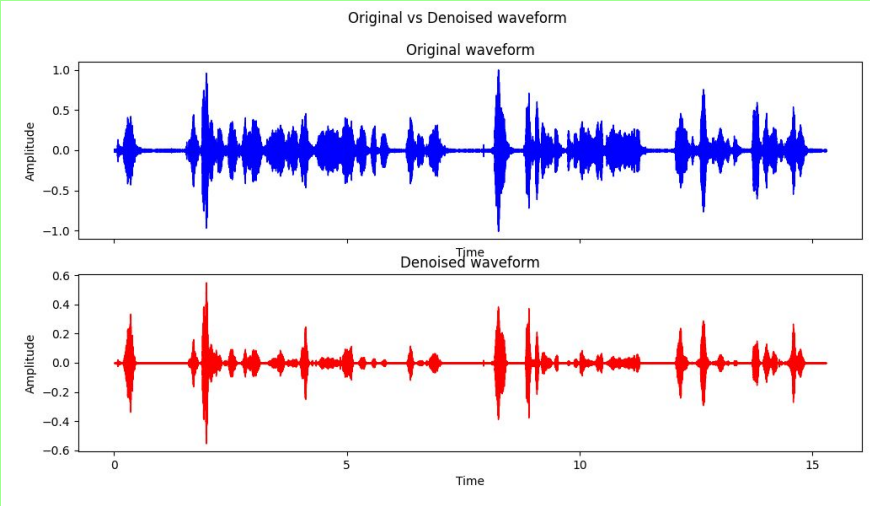
Extracted information:

- Number of words in the audio.
- Number of words per second.
- Tempo
- Fundamental frequency:

  - Mean
  - Median
  - Standard deviation
  - 5-percentile
  - 95-percentile
  - Standard deviation

Preprocessing steps are necessary:

DENOISING: Performed using the Noisereduce library.

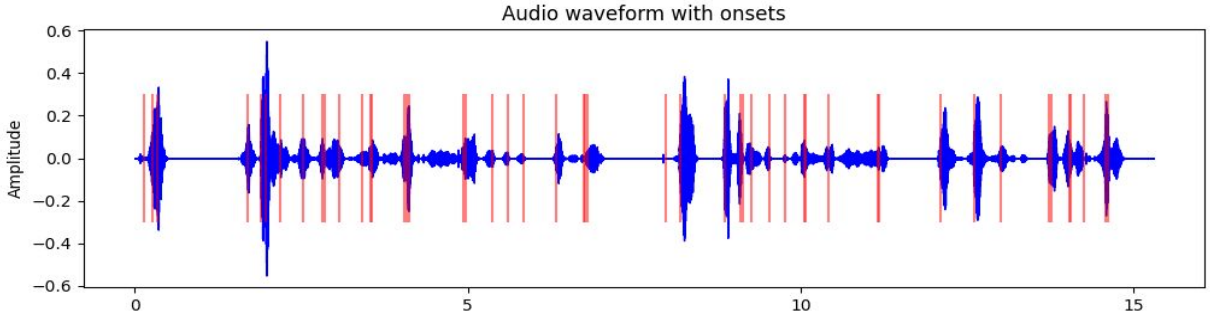


TRIMMING: ✗

Not implemented, **because we consider that the pause or the urge before starting to talk is a characteristic of the age of the speaker.**

Task e. Define a strategy to represent acoustic data

By finding the ***onset*** we compute the ***number of spoken words per second.***
With some exceptions, but we expect younger people to have higher values of ***number of words per second*** and ***beats per minute.***

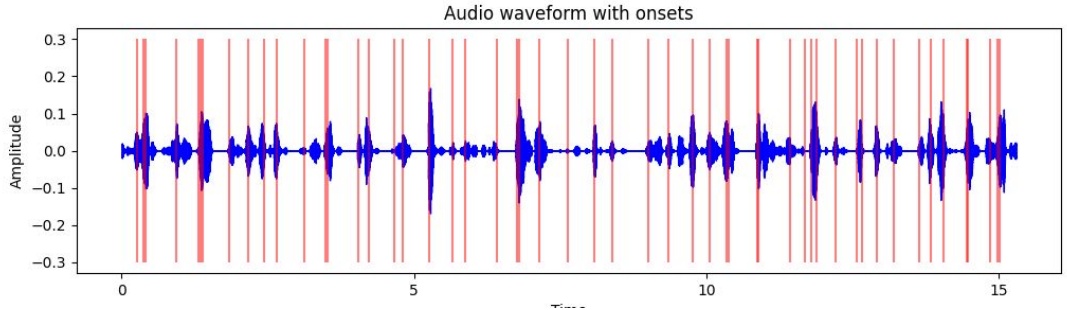


Age class: 6 → 46-60 years

Audio length: 15.302s
Number of words/s: 3.20
Beats per minute (bpm): 18.38

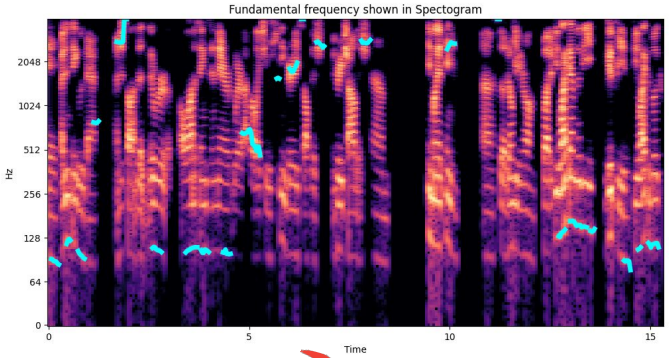
Age class: 2 → 14-18 years

Audio length: 15.302s
Number of words/s: 3.46
Beats per minute (bpm): 21.31



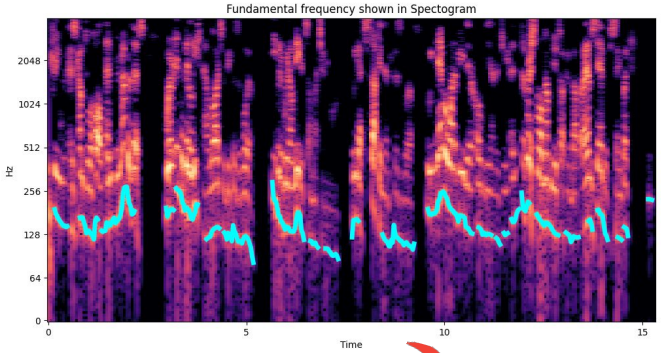
Task e. Define a strategy to represent acoustic data

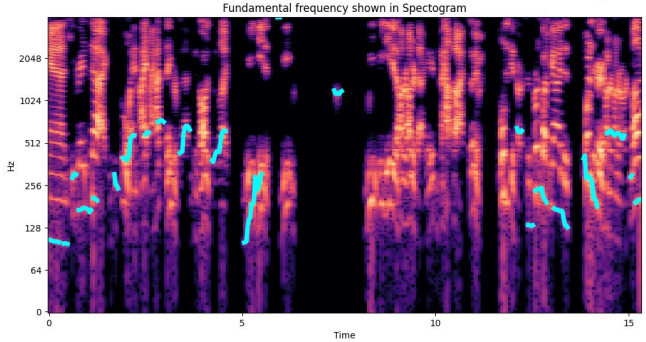
Fundamental frequency: Lowest frequency at which a periodic sound appears (**pitch**).



Age class: 1 → 7-13 years

f0 mean:	443.26
f0 median:	292.60
f0 standard deviation:	544.91
f0 5-percentile:	102.56
f0 95-percentile:	1145.30





Age class: 4 → 25-32 years

f0 mean:	971.51
f0 median:	151.46
f0 standard deviation:	1242.97
f0 5-percentile:	89.64
f0 95-percentile:	3993.93

Age class: 7 → 61+ years

f0 mean:	154.15
f0 median:	145.46
f0 standard deviation:	39.86
f0 5-percentile:	104.68
f0 95-percentile:	233.51

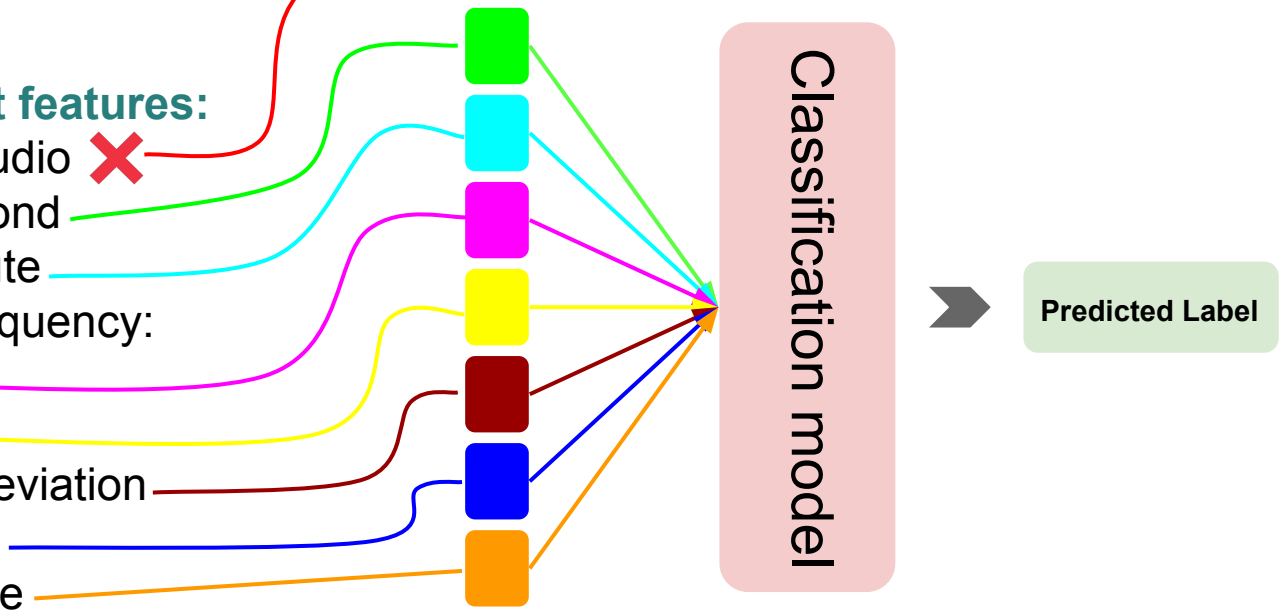
Task e. Define a strategy to represent acoustic data

Audio pipeline scheme:

1st step: Preprocess data:
- Divide data by 2^15
- Noise reduction

2nd step: Extract features:
- # words in the audio ✗
- # words per second
- # beats per minute
- Fundamental frequency:
  - Mean
  - Median
  - Standard Deviation
  - 5-percentile
  - 95-percentile

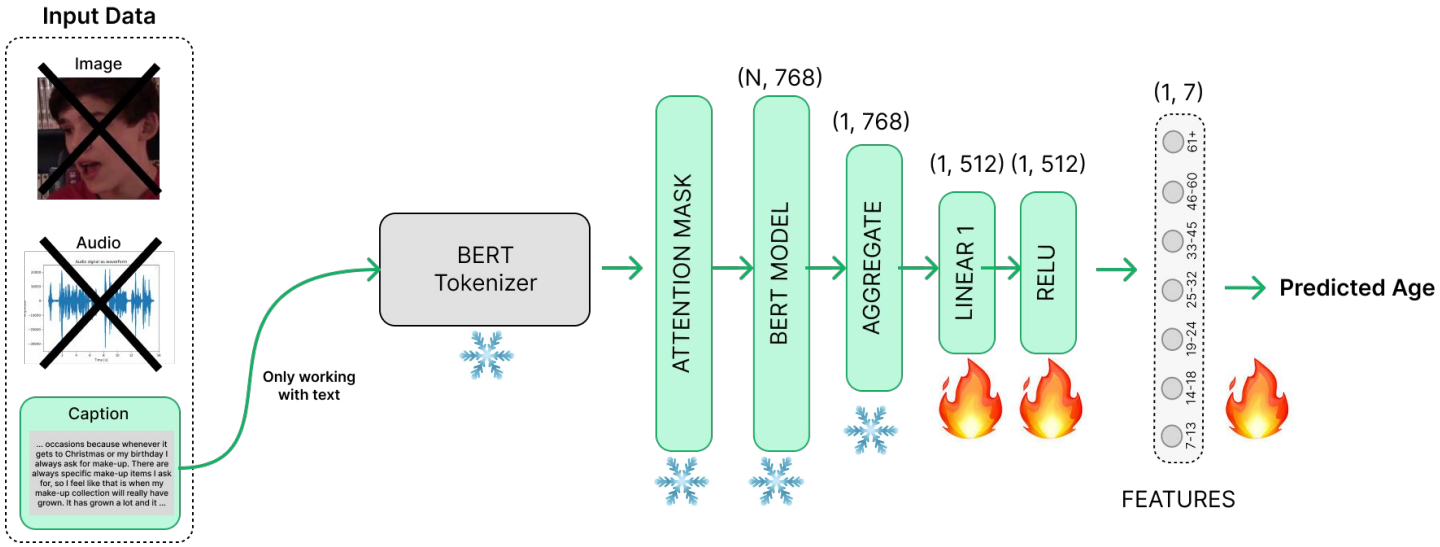
The number of words in the audio depends on the sentence itself, **it is a characteristic of the text, not from the speaker**.



Task f. Define a strategy to represent text data

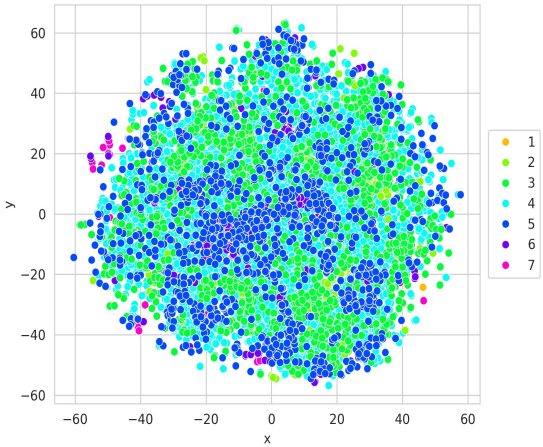
Task f. Define a strategy to represent text data

As a textual encoder to extract features from text, we've used BERT from the HuggingFace transformer's library.
Afterwards we added 2 non-linear layers at the top with ReLU activations.

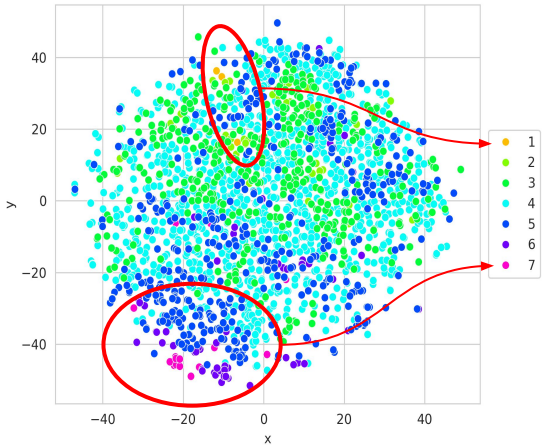


Task f. Define a strategy to represent text data

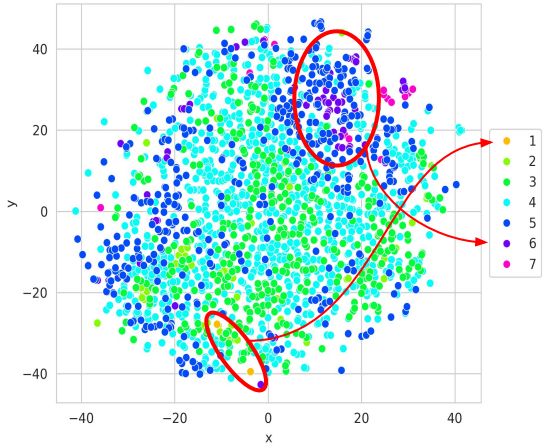
Train TSNE



Val TSNE



Test TSNE

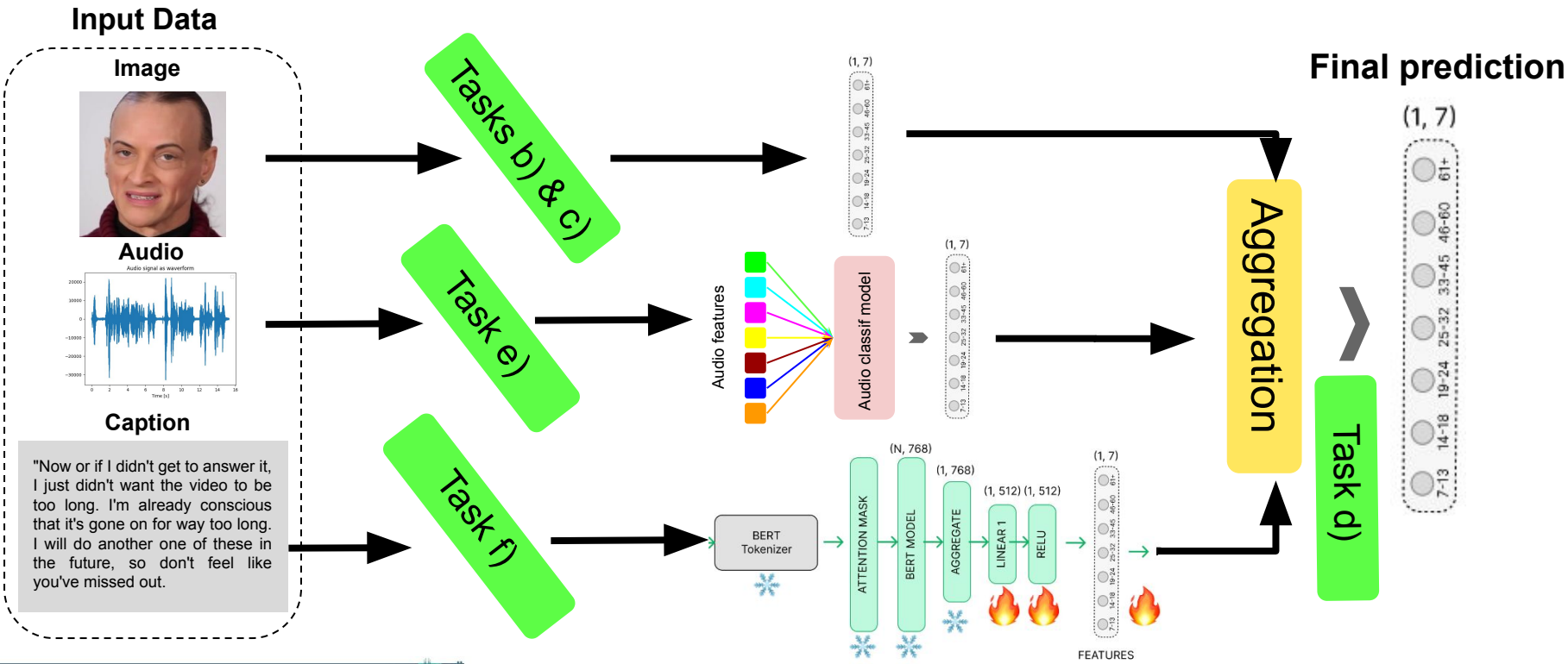


All classes spreaded out. But, for some very specific age intervals, such as old or very young people (classes 1→ [7, 13], 6 → [46, 60], 7 → 61+), their embeddings are generally more concentrated in specific locations.
So, **we consider important for just some specific age intervals.**

Task g. Multimodal model

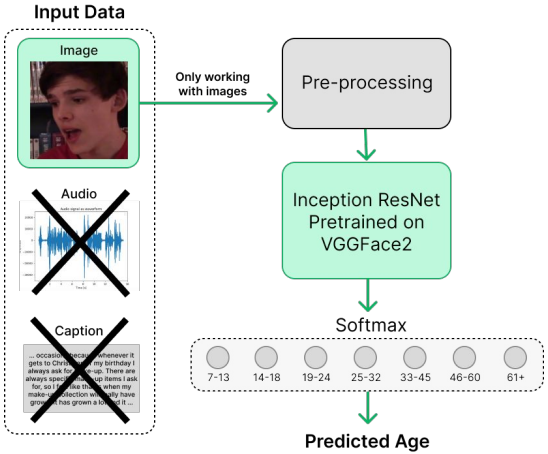
Task g. Multimodal model

STRATEGY: **Late fusion.** We have trained each model separately and combined them afterwards.



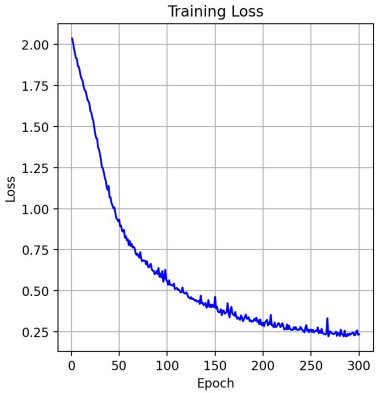
Task g. Multimodal model

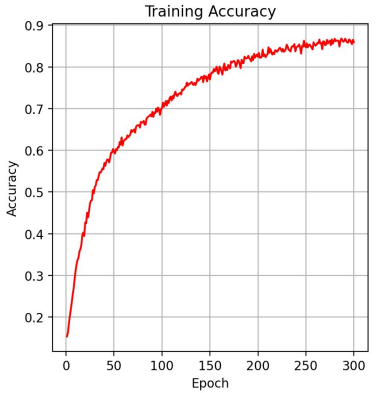
Image Classification Model:

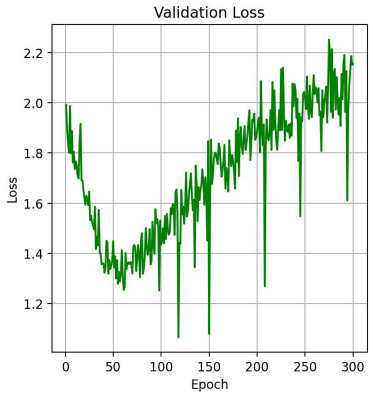


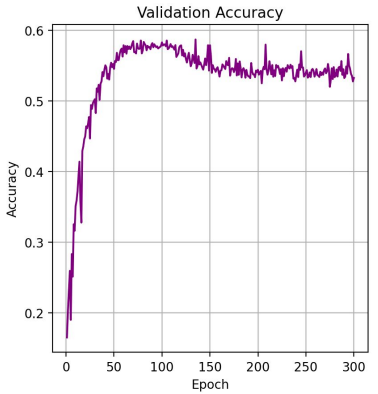
Loss: Weighted Cross Entropy
Optimizer: AdamW
Learning Rate: 1e-6
Batch Size: 256

Image model Test Accuracy: **0.54**









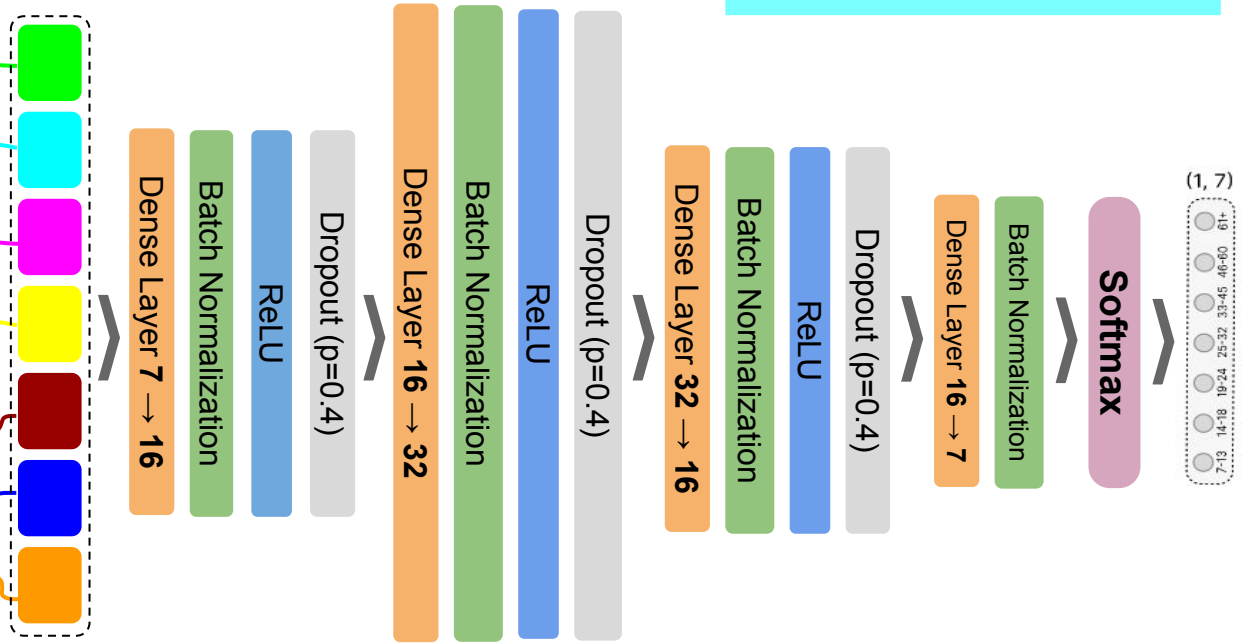
Task g. Multimodal model

Audio Classification Model:

Loss: Weighted Cross Entropy
Optimizer: AdamW
Learning Rate: 1e-6
Batch Size: 32

TIME COMPARISON:
- **Before new strategy:**
~18s per audio,
~31 hours per epoch
- **After new strategy:**
~2.4s per audio,
~16 minutes per *(reduced)* epoch

- # words per second
- # beats per minute
- Fundamental frequency:
  - Mean
  - Median
  - Standard Deviation
  - 5-percentile
  - 95-percentile

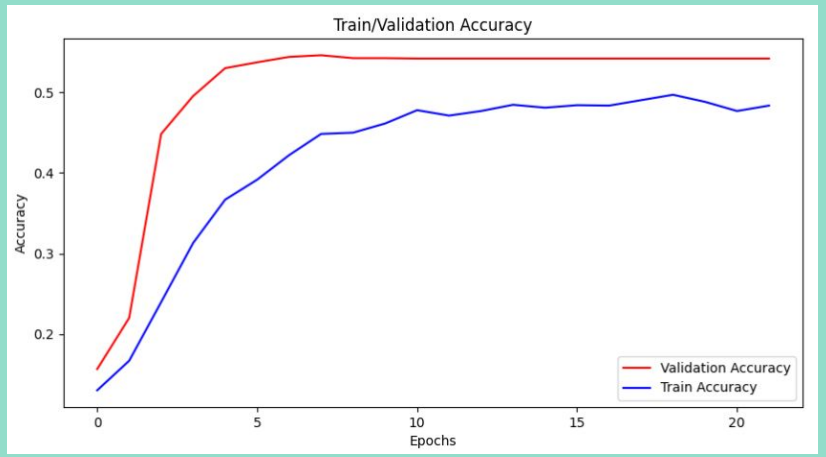


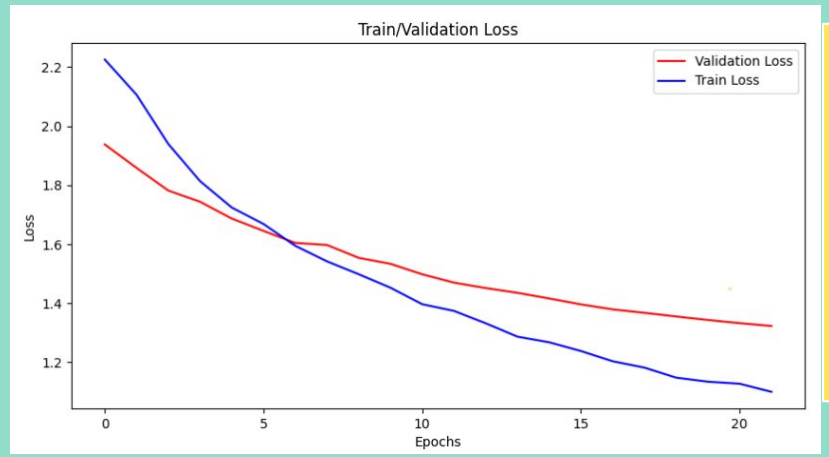
Task g. Multimodal model

Audio Classification Model:

Loss: Weighted Cross Entropy
Optimizer: AdamW
Learning Rate: 1e-6
Batch Size: 32

TIME COMPARISON:
- **Before new strategy:**
~18s per audio,
~31 hours per epoch
- **After new strategy:**
~2.4s per audio,
~16 minutes per *(reduced)* epoch





Audio model test accuracy:

49.37%

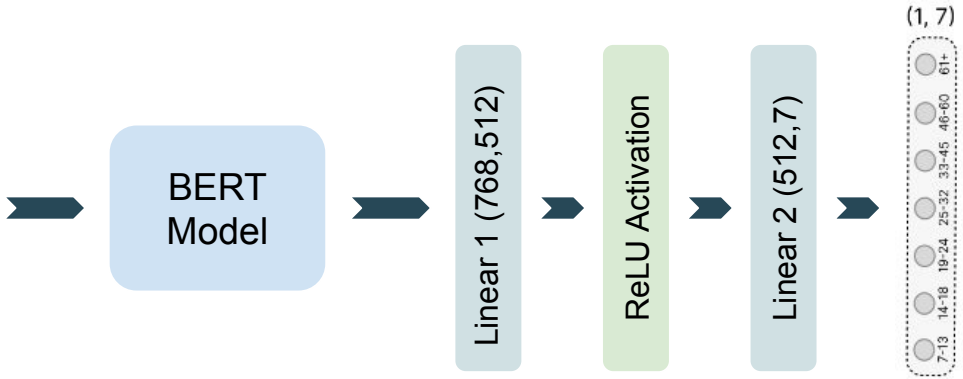
- 95-percentile

Task g. Multimodal model

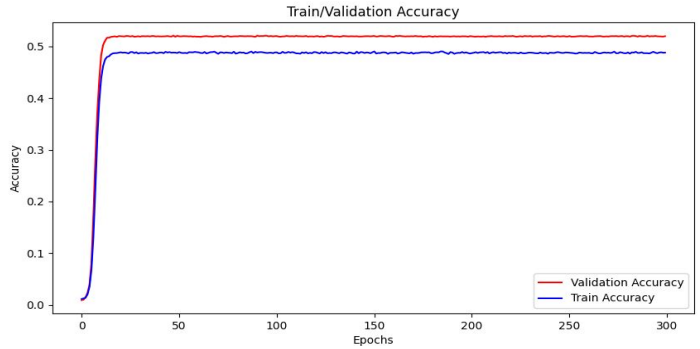
Text Classification Model:

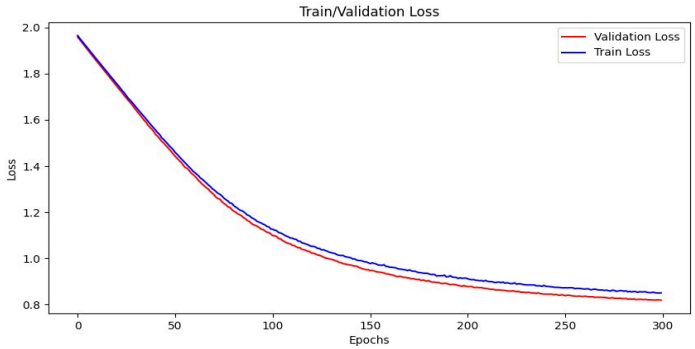
Loss: Weighted Cross Entropy
Optimizer: AdamW
Learning Rate: 1e-6
Batch Size: 256

"Now or if I didn't get to answer it, I just didn't want the video to be too long. I'm already conscious that it's gone on for way too long. I will do another one of these in the future, so don't feel like you've missed out.



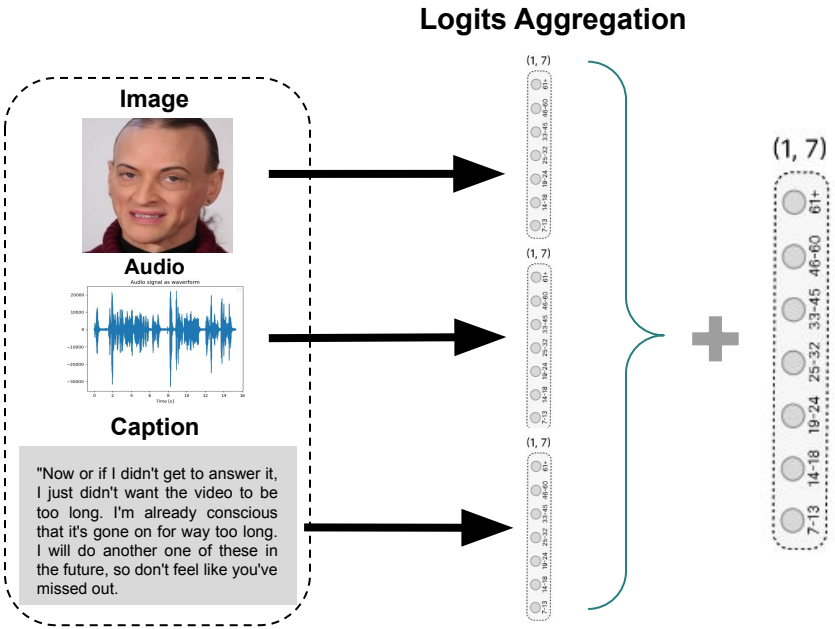
Text model Test Accuracy: **0.49**





Task g. Multimodal model

Combining Models:



Predicted Age

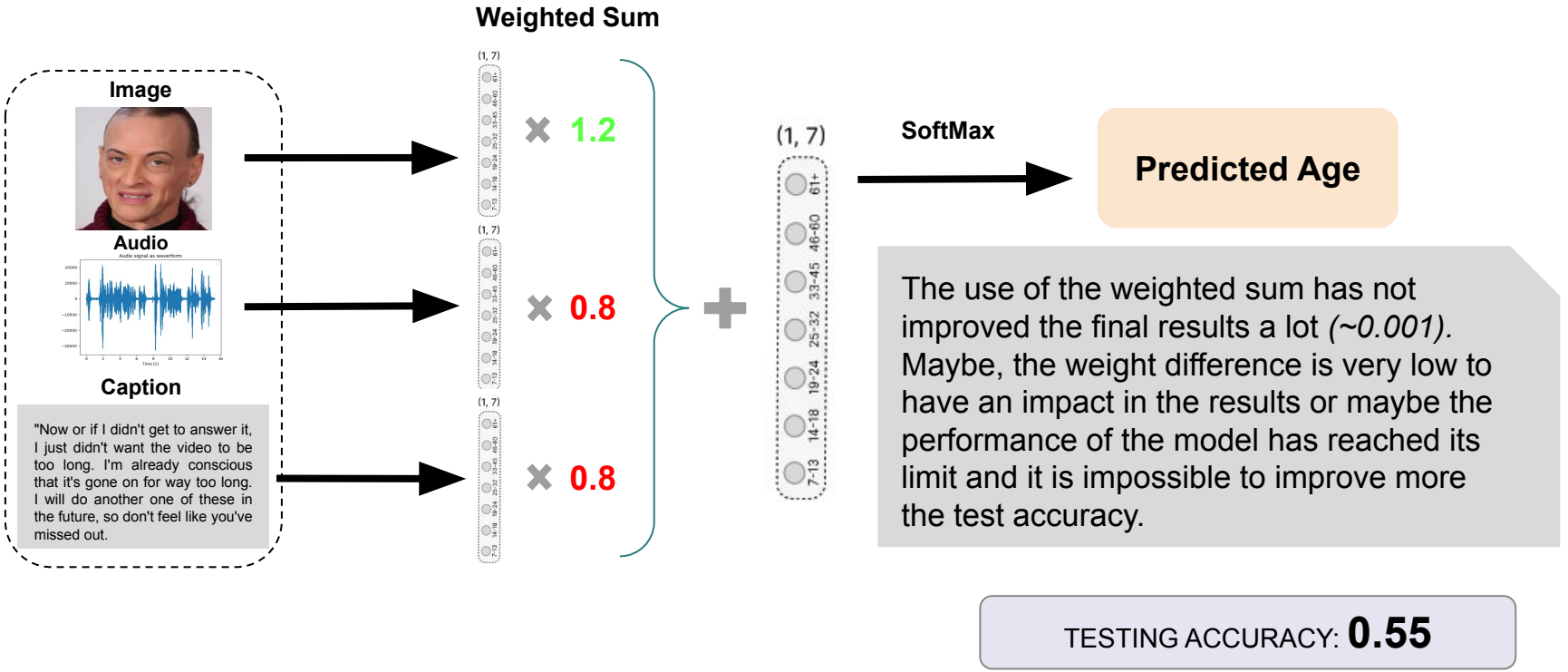
With the combination of predictions we have obtained an improved solution in terms of Test Accuracy. **We have outscored the results of task b** only using images, so we can confirm that **the use of different data types in a multimodal model is a way of improve our classification model.**

Image Test Accuracy: **0.535**
Audio Test Accuracy: **0.49**
Text Test Accuracy: **0.49**

COMBINED TESTING ACCURACY: **0.55**

Task h. Ablation Study

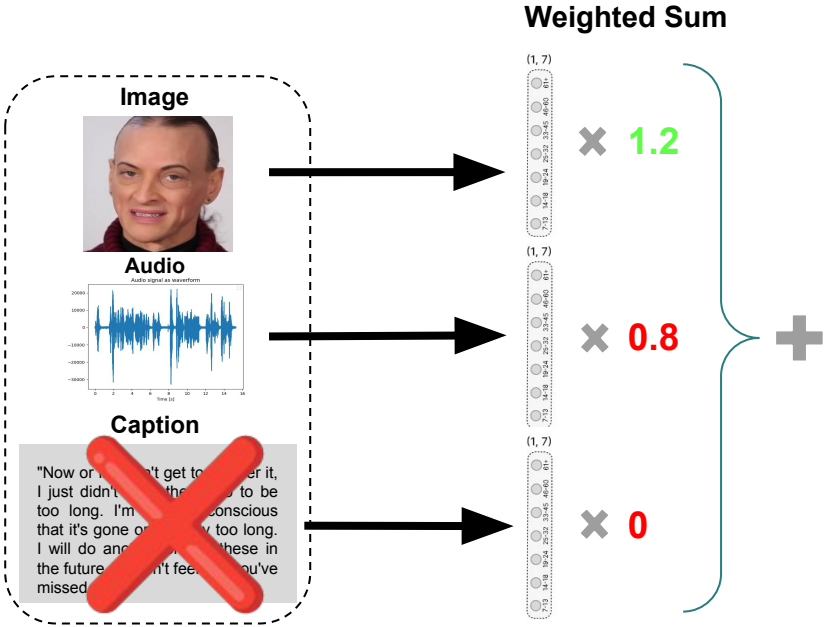
Task h. Ablation Study



The use of the weighted sum has not improved the final results a lot *(~0.001)*. Maybe, the weight difference is very low to have an impact in the results or maybe the performance of the model has reached its limit and it is impossible to improve more the test accuracy.

TESTING ACCURACY: **0.55**

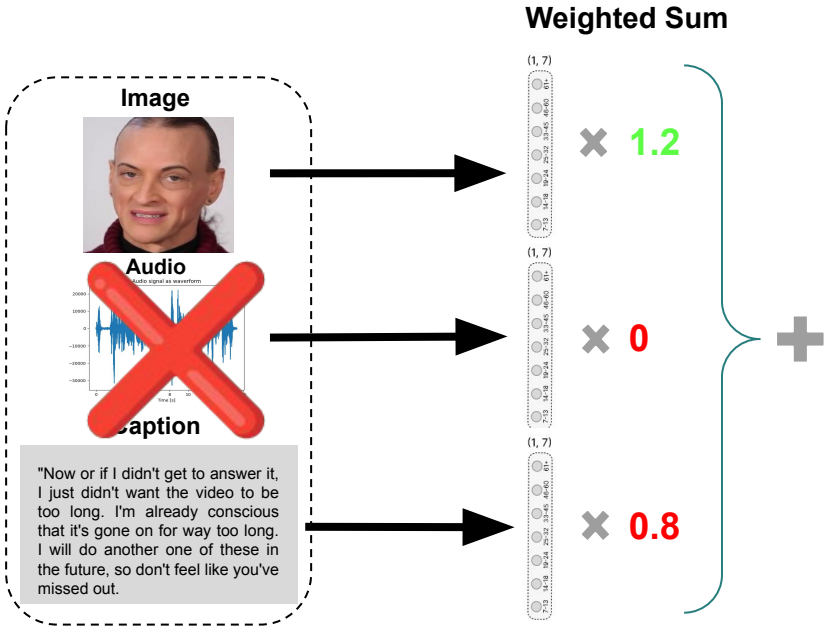
Task h. Ablation Study



After proving this new method the results are a bit lower than the previous one using the text. So, the text does have a positive influence on the results and it is important to include it in the multimodal model implementation.

TESTING ACCURACY: **0.54**

Task h. Ablation Study



Again, the combination of only two data types gives us lower results than the full combination. In this case the results are a bit lower than only image + audio, so the importance of the audio data is higher than the text, confirming what we expected.

TESTING ACCURACY: **0.54**

Task i. Conclusions

Task i. Conclusions

- The unbalance of the dataset is a big problem in order to obtain better results. We have tried to reduce the negative effect through data-augmentation but the extremely difference between the number of some classes prevents us from obtaining higher results.

- The combination of different data sources in the multimodal model provides us an improvement in the final accuracy results. Moreover after the ablation study we have confirmed the importance of using the three data types, and the higher importance of audio over text.

- Finally, our late fusion model is very limited by the individual model results, so in future implementations we should combinate the different data before training an unique model.