Data

# How all start ?

**178 images**

with bbox of a calf face and health state (Diarrhea, Pneumonia or Healthy)

+

9.622 videos

Of 1h max from 4 channels

+

1.829 lignes

Of calf health assessments covering 41 days

+

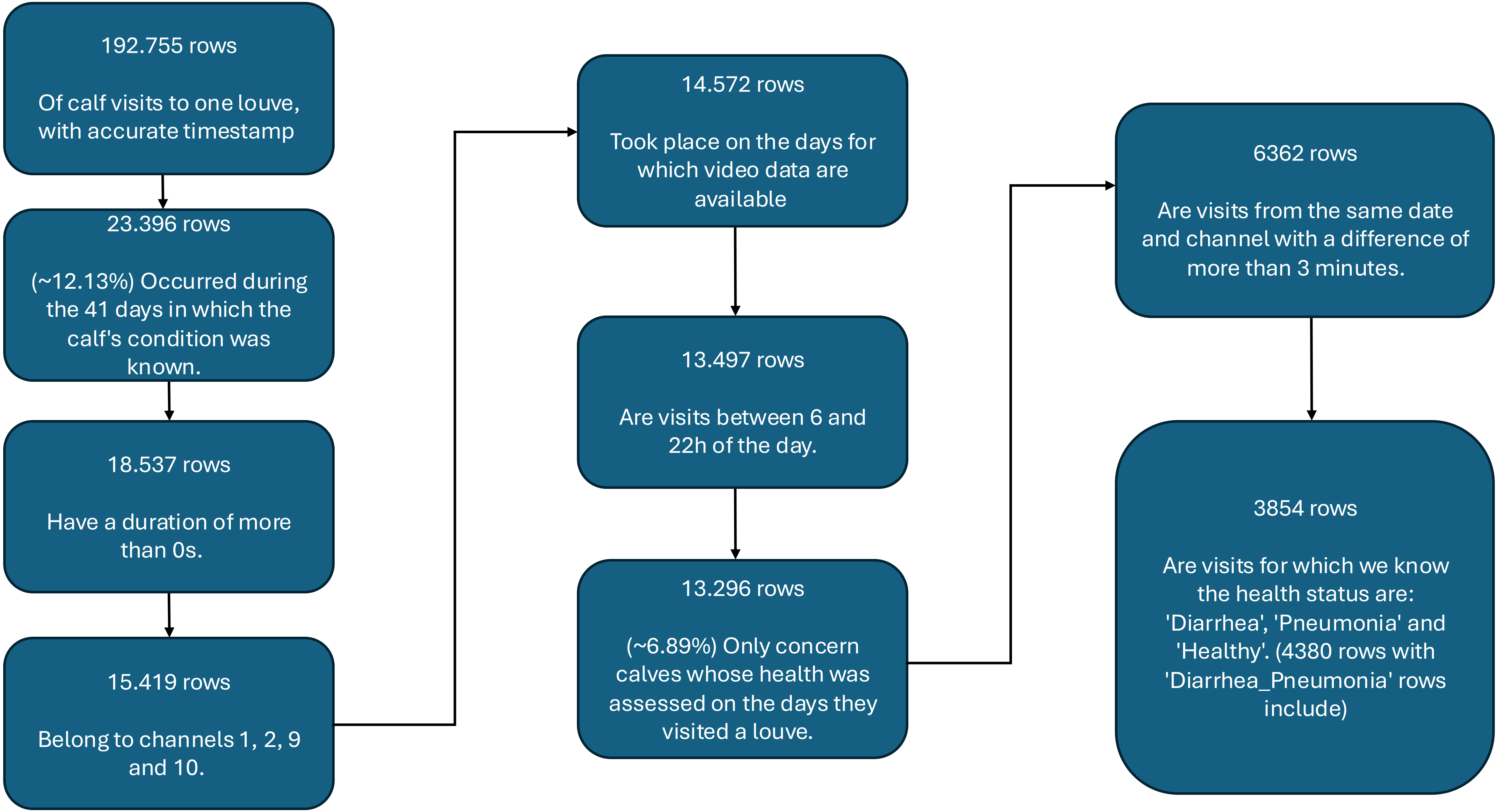
192.755 lignes

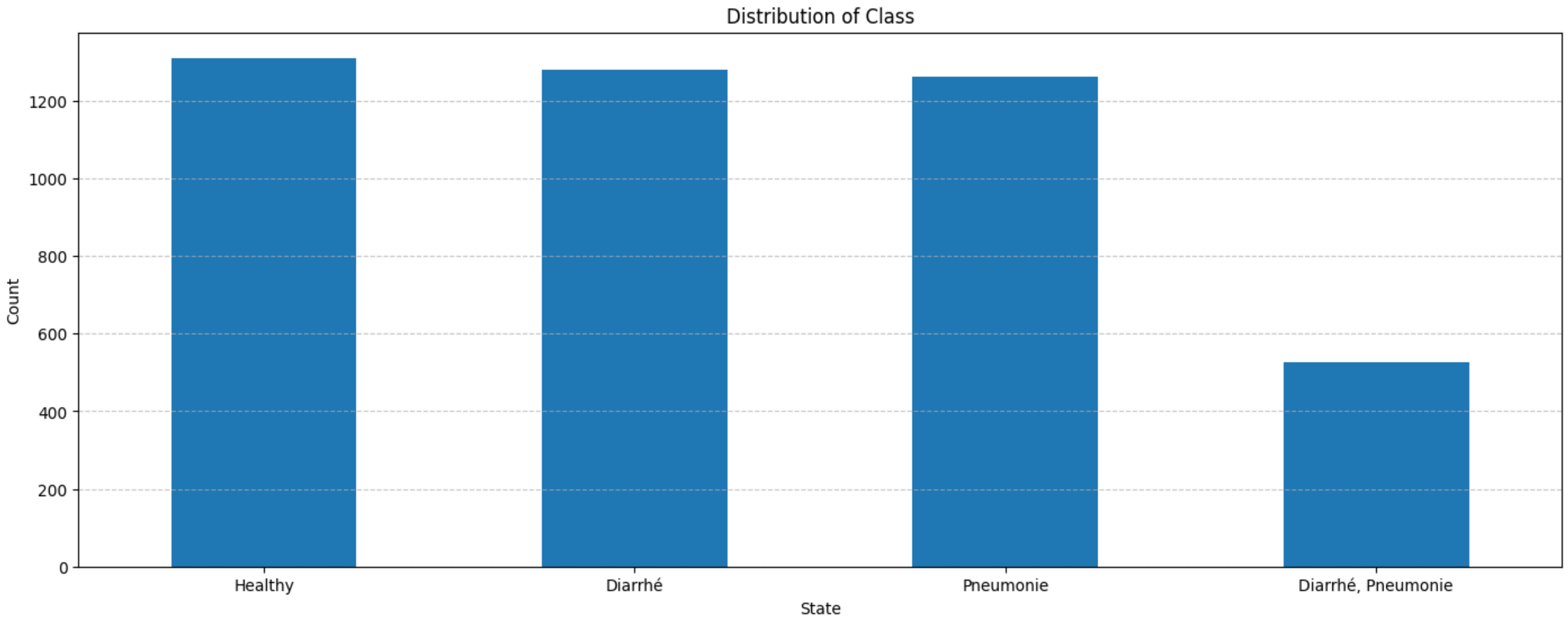
Of calf visits to one louve, with accurate timestamp

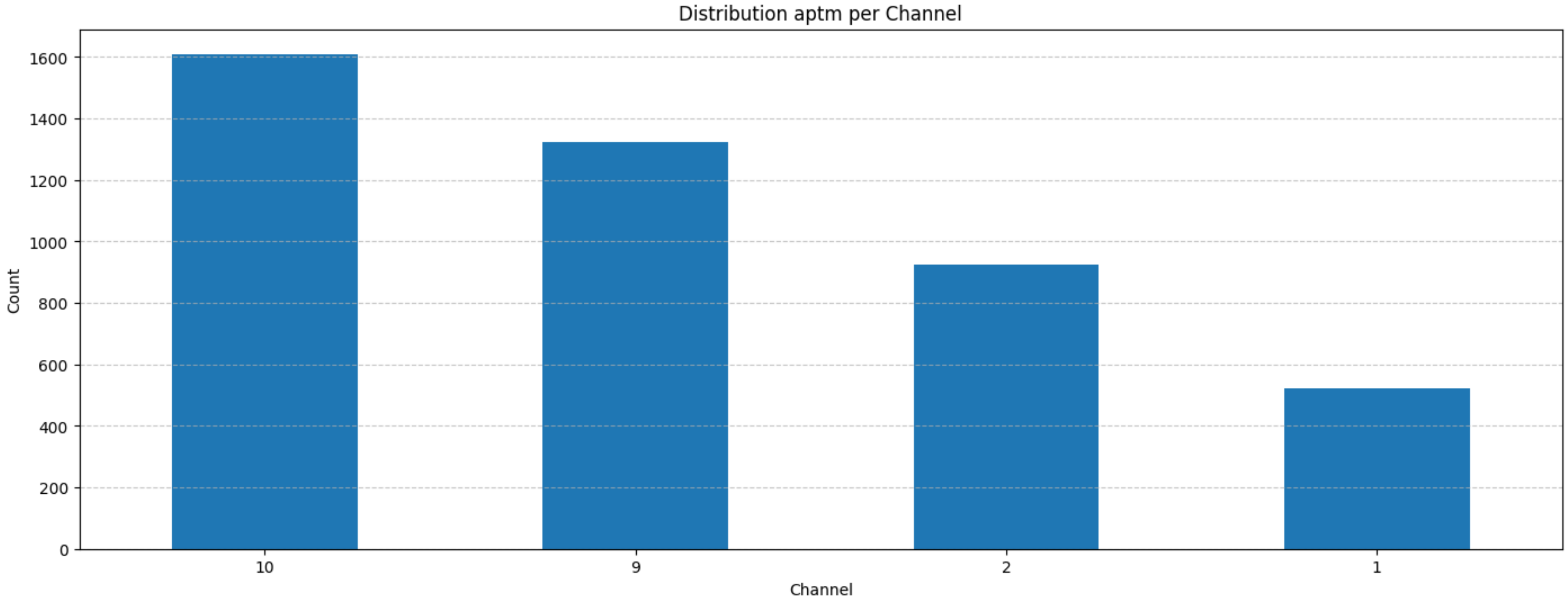
Our Goal:

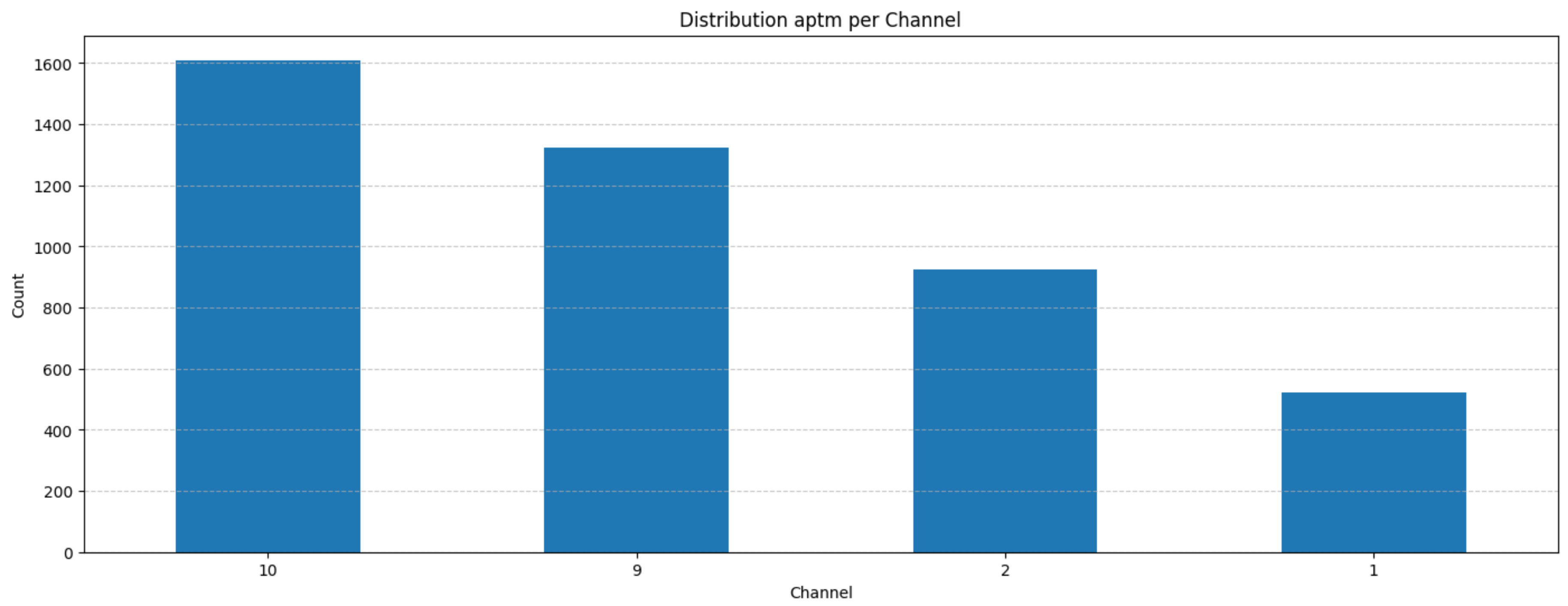
Infer calf health state from calf images/videos

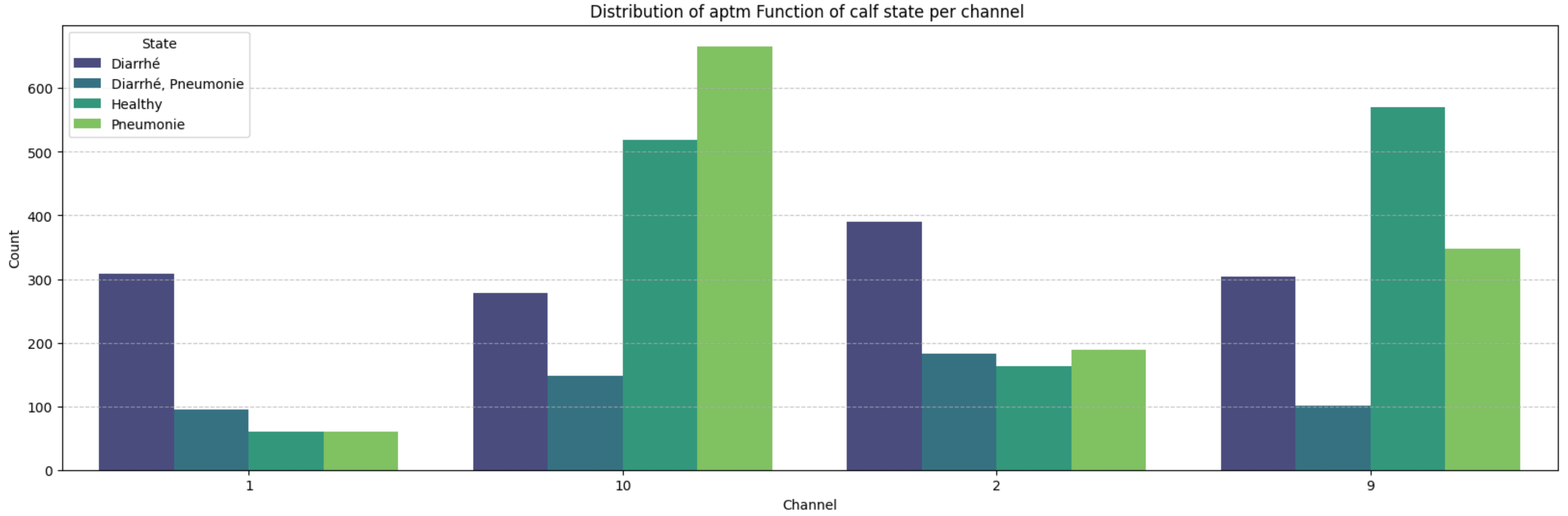
We need more data, but how do we get them?




Distribution of Class
Count
0
200
400
600
800
1000
1200
Healthy
Diarrhé
Pneumonie
Diarrhé, Pneumonie
State


Distribution aptm per Channel
Count
0
200
400
600
800
1000
1200
1400
1600
10
9
2
1
Channel


Distribution aptm per Channel
Count
Channel
10
9
2
1
0
200
400
600
800
1000
1200
1400
1600


Distribution of aptm Function of calf state per channel
State
Diarrhé
Diarrhé, Pneumonie
Healthy
Pneumonie
Count
Channel
1
10
2
9
0
100
200
300
400
500
600

# We need automatically anotate data : From Yolo 0 to Yolo Last

Yolo 0 training details

- 178 images as dataset
- Default data augmentation
- Train on 5 epochs
- Task: detect a calf face
- mAP50: 0.803

Yolo Last training details

- 178 images as dataset
- Apply custom data augmentation on the images (GaussianBlur, MedianBlur, Sharpen, Flip, Rotation between (10, 20)deg )
- Generate new 890 images
- Train on 5 epochs
- Same task as before
- mAP50: 0.975

Yolo 0

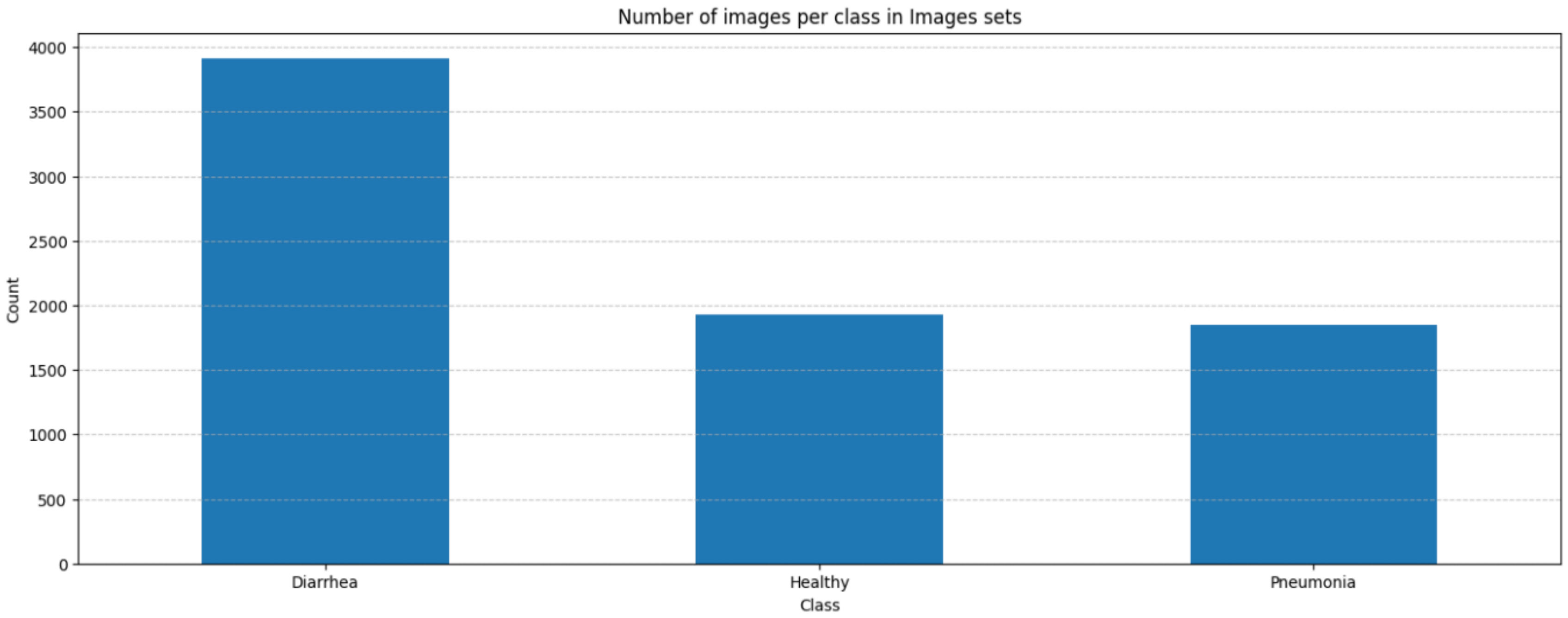

Yolo last


Yolo 0

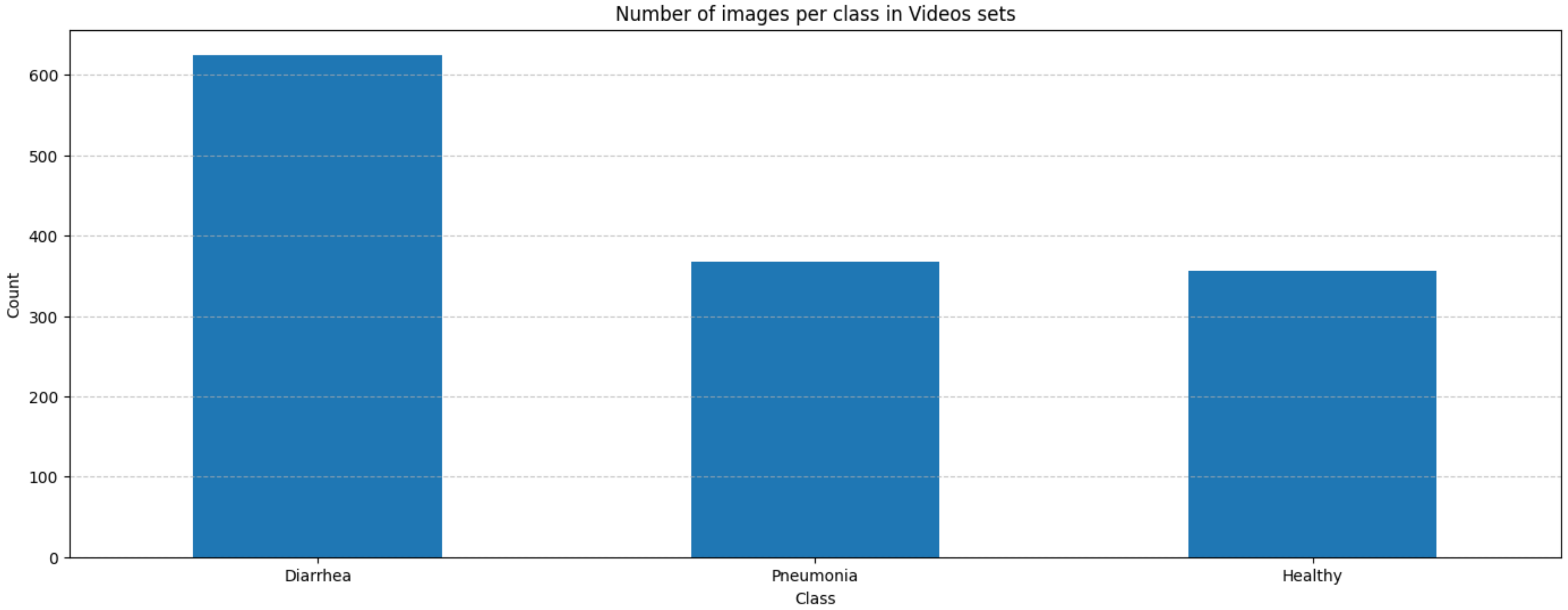

Yolo last

- **How Yolo Last help ?**
  - Sample 30 frames evenly from a 10s videos just before the calf start eating,
  - Use the model to detect at least one face on each frame, at .80 of confidence,
  - Save the video where it detect at least one time,
  - And images where it detect the calf face.

- **The results from** 3854 visits**:**
  - 1349 Videos
  - 7687 Images
  - With 76 unique calf
  - With 37 unseen before by the model




- **The results:**
  - 1349 Videos
  - 7687 Images
  - With 76 unique calf
  - With 37 unseen before by the model



# Training Data

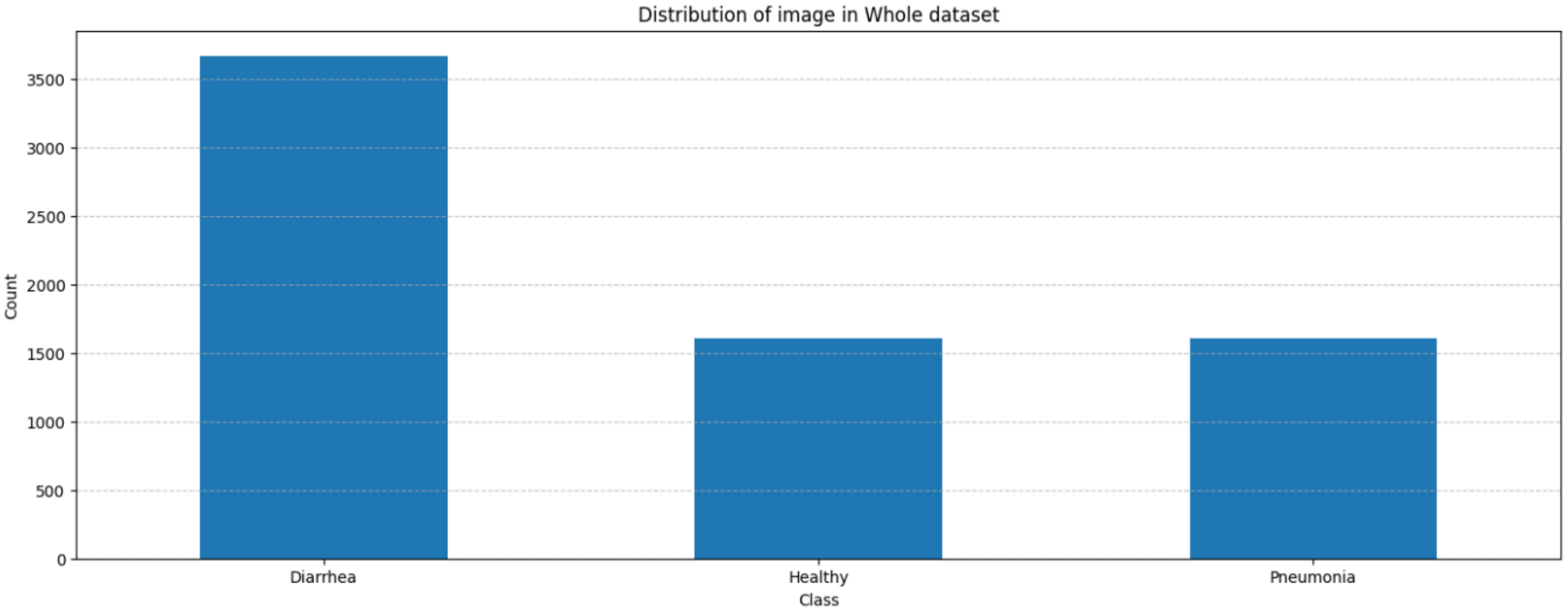
# Videos Training set details

| Videos | Sample | Whole |
|---|---|---|
| Training set | 207 | 972 (80% of 1215) |
| Validation set | 1008 (only use 10-20%) | 243 (20% of 1215) |
| Test set | 68 | |
| Calf number in Train + Val set | 44 | |
| Calf number in Test set | 24 | |

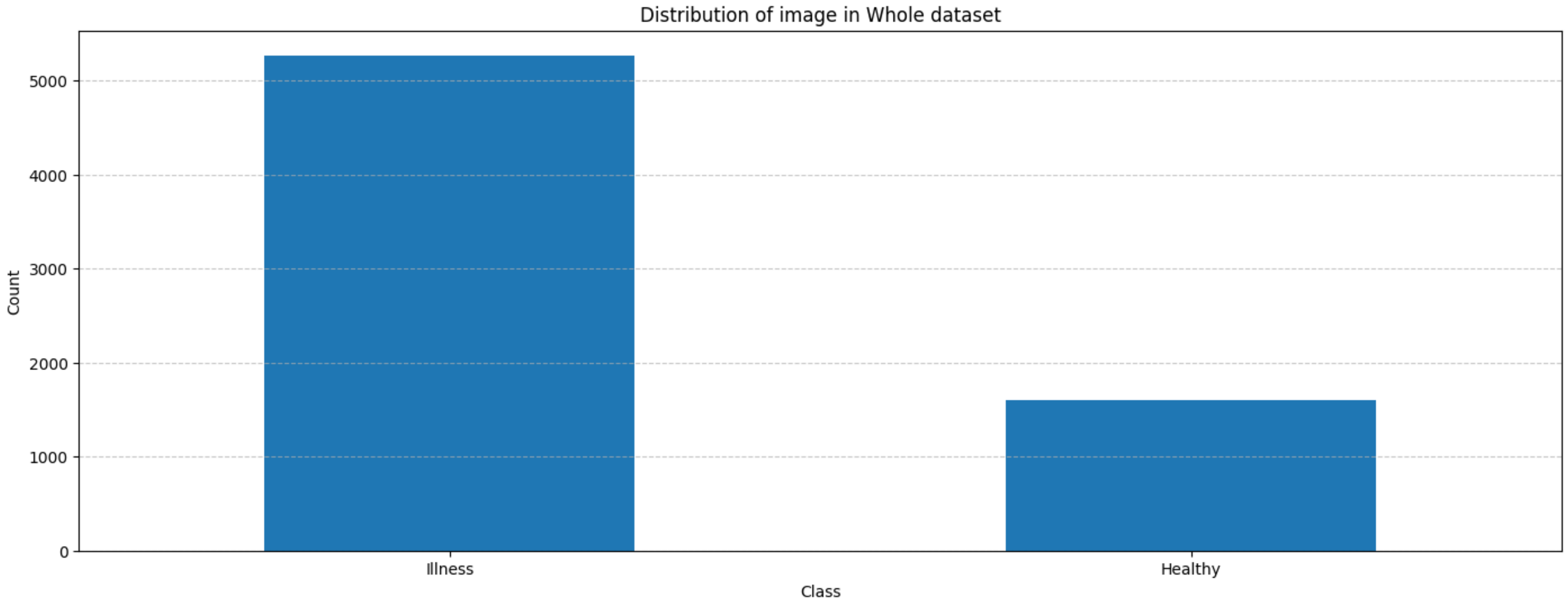
# Images Training set details

| Images | Sample | All |
|---|---|---|
| Training set | 213 | 5720 (80% of 7149) |
| Validation set | 6936 (only use 10-20%) | 1429 (20% of 7149) |
| Test set | 283 from videos test set | |
| Calf number in Train + Val set | 44 | |
| Calf number in Test set | 24 | |

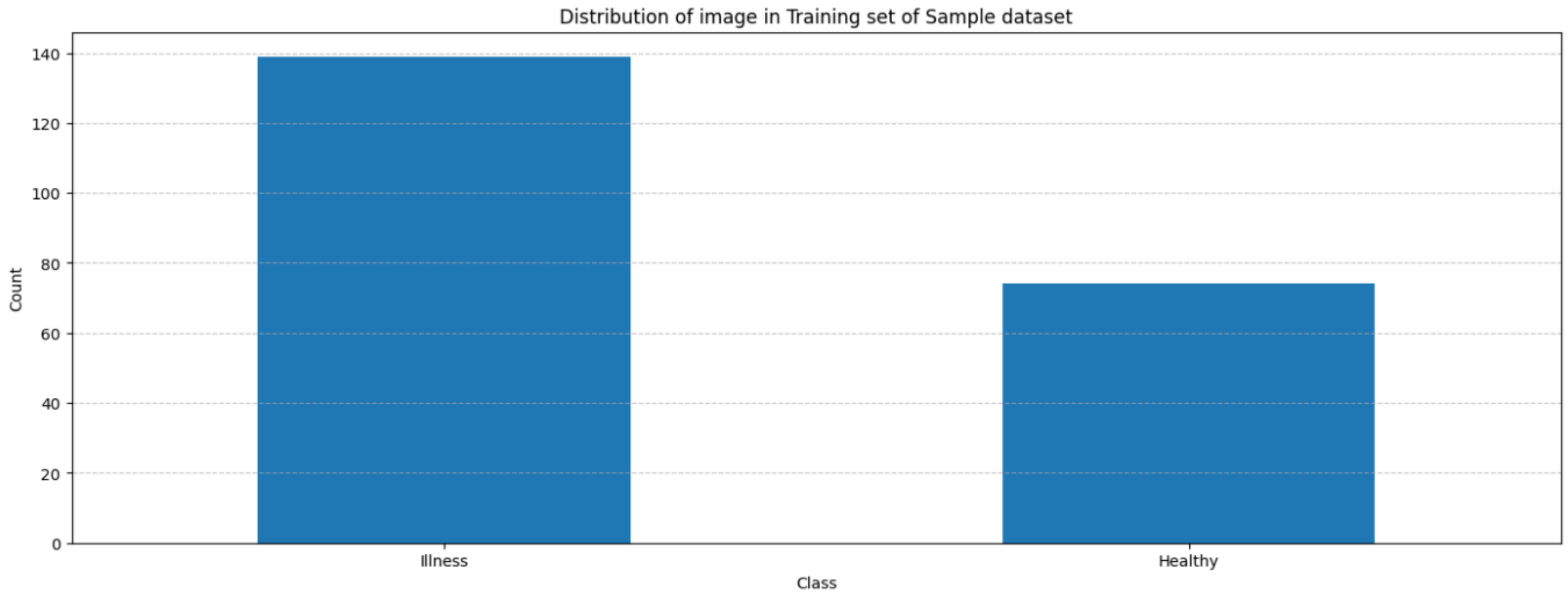
# Training details



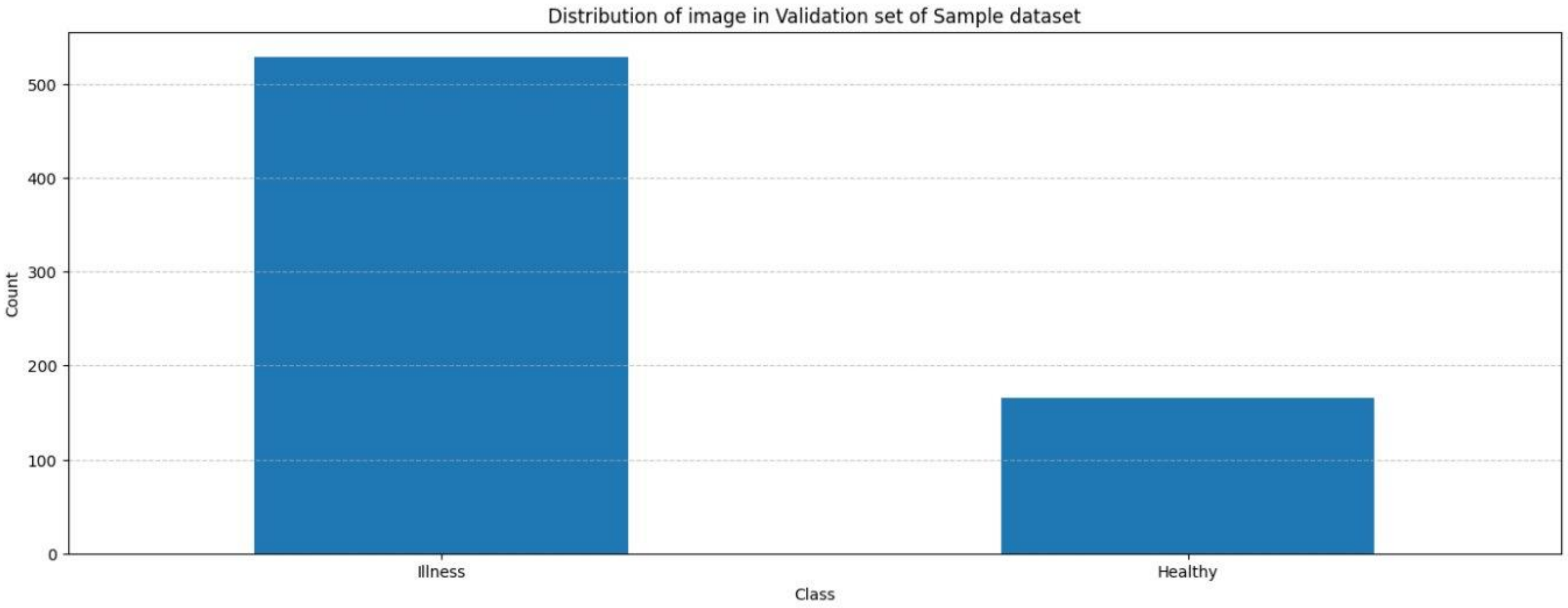

- Do a train-test split at 20%
- Then use 80% in Training set
- The other part for validation set



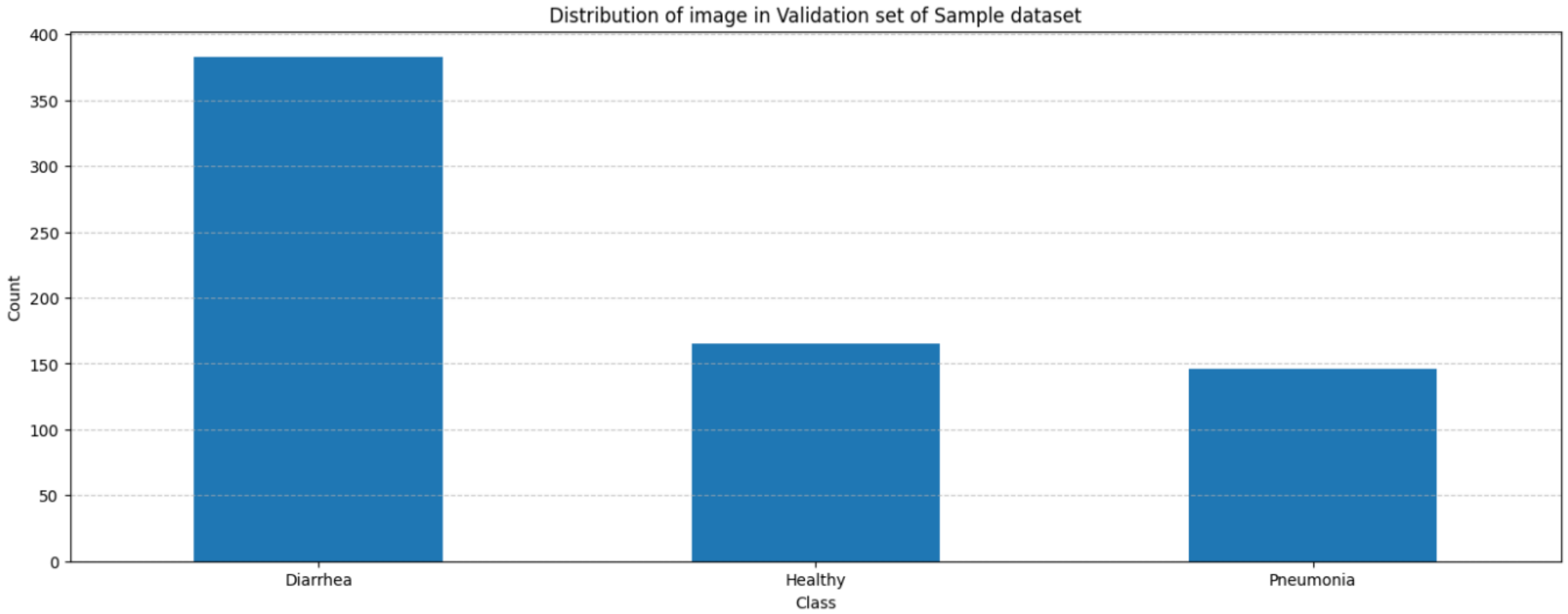
# Training details

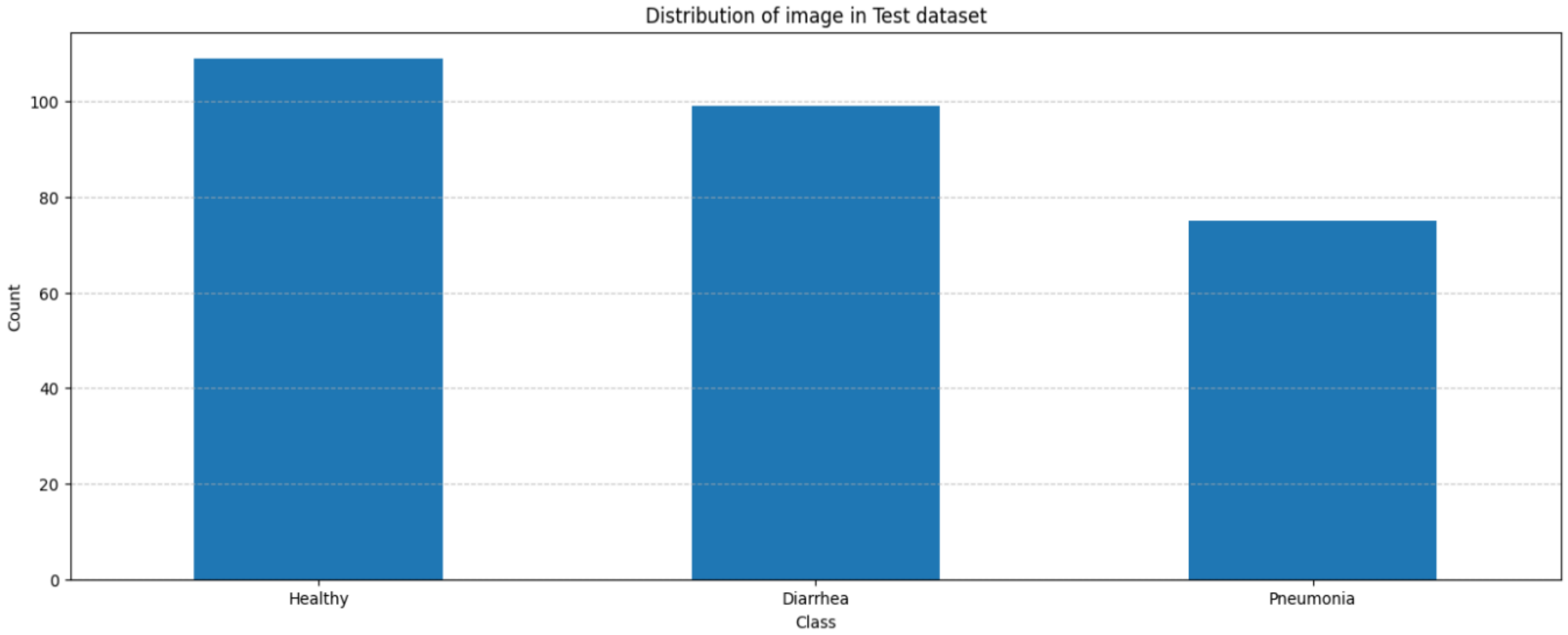
# Training details



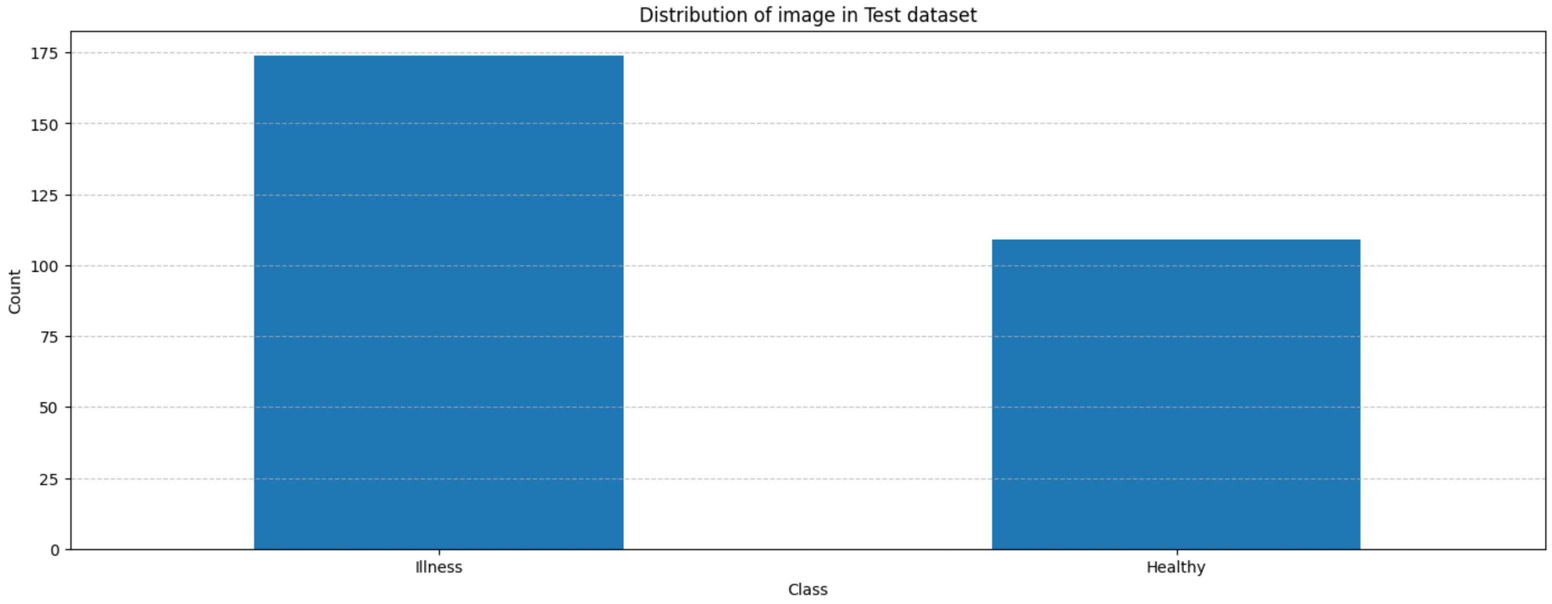

- Use only 10-20% of this set for validation



# Training details




- This distribution almost same for video sets



# How I train each model ?

Image models

- Used a pretrained model
- Balance each batch
- Used a weighted loss
- Train over 10 epochs with early stop
- Test on the best model base on lower loss on training

Video models

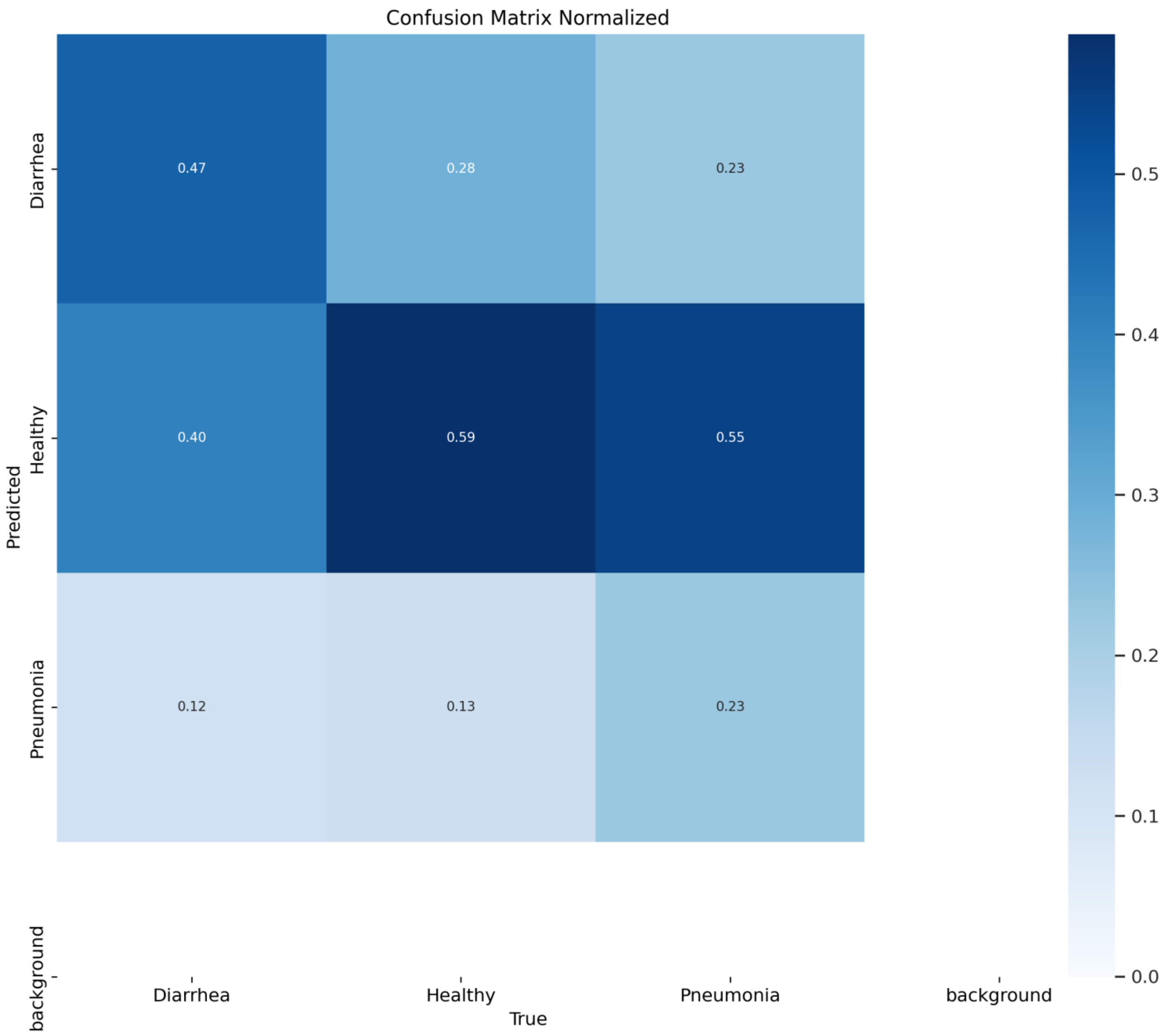
- Used a pretrained model
- 10s of videos
- 16 frames per videos
- Balance each batch
- Used a weighted loss
- Train over 10 epochs

# Training Results

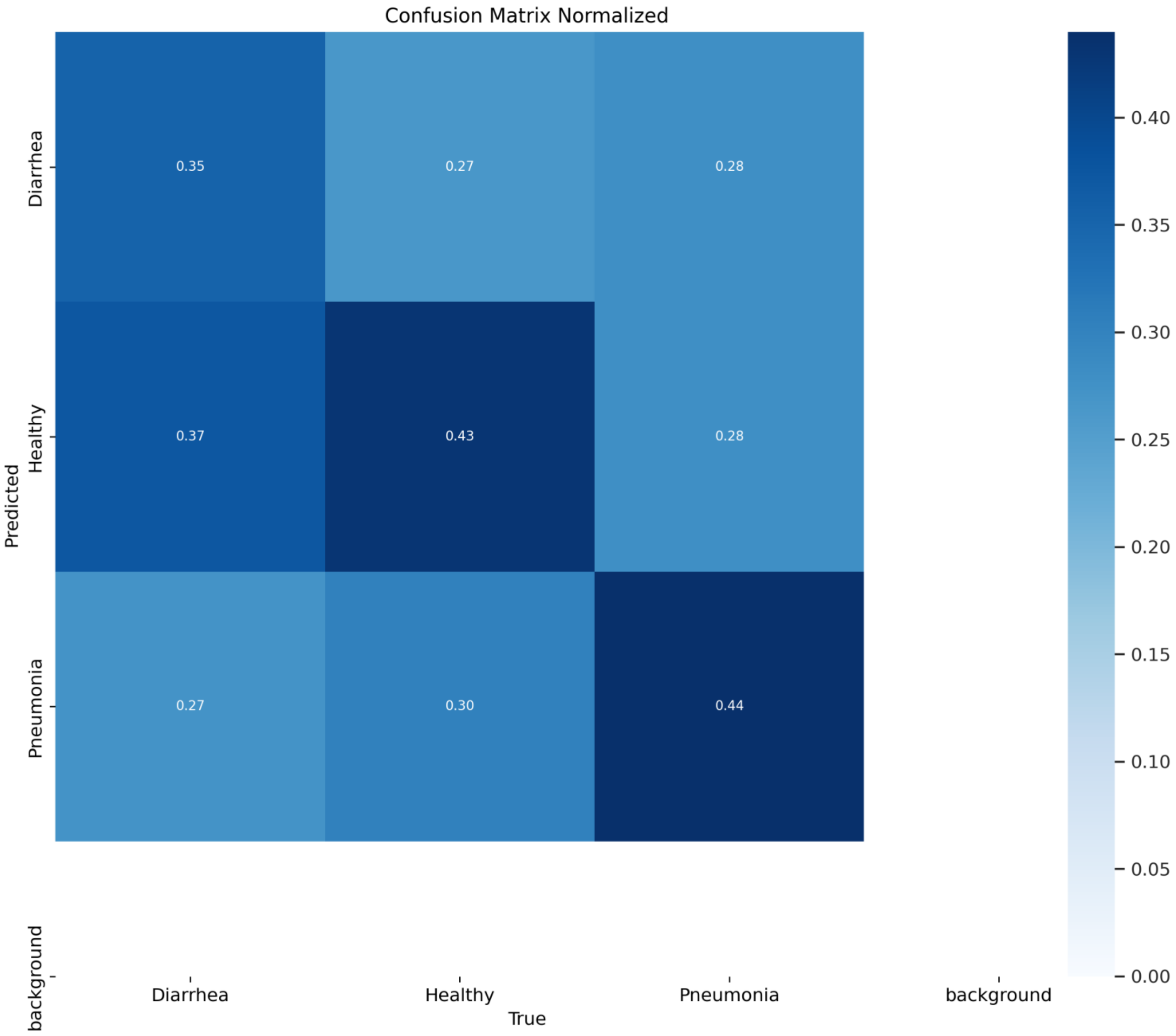
## How good they perform ?

| Images Models (Value in %) | Accuracy | | | | F1-score | | | | Binary Accuracy | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Two Class: | | Three class: | | Two Class: | | Three class: | | Two Class: | | Three class: | |
| | Sample | Whole | Sample | Whole | Sample | Whole | Sample | Whole | Sample | Whole | Sample | Whole |
| ViT | 46.29 | **63.60** | 39.22 | 43.11 | 54.76 | **58.63** | 31.21 | 42.81 | 53.41 | **64.23** | 38.52 | 42.89 |
| InceptionV3 | 41.70 | **53.36** | 34.28 | 40.99 | **56.69** | 51.47 | 32.72 | 39.20 | 52.41 | **55.39** | 33.56 | 38.02 |
| Efficientnet-b3 | **53.71** | 49.82 | 42.40 | 40.64 | **58.93** | 56.17 | 40.39 | 40.11 | **59.79** | 56.11 | 39.66 | 39.05 |
| INTR | 44.16 | **55.12** | 40.98 | 42.40 | 41.48 | **60.92** | 38.31 | 42.64 | 45.51 | **61.79** | **61.79** | 42.98 |
| Yolov8 | - | - | 40.6 | 45.2 | - | **-** | - | - | - | **-** | - | - |

# Yolo confusion matrix on Whole set



# Yolo confusion matrix on Sample set

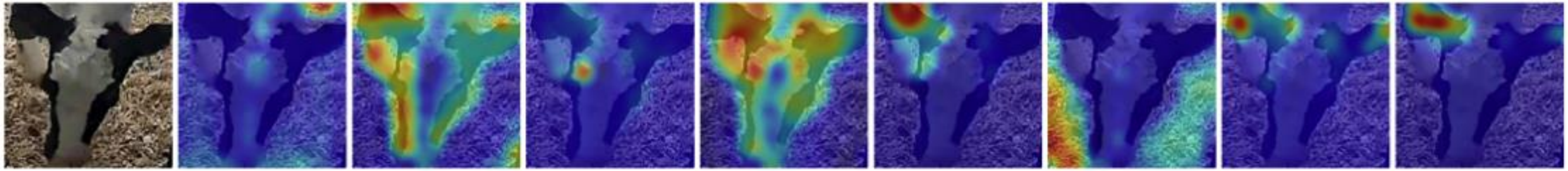


# How good they perform ?

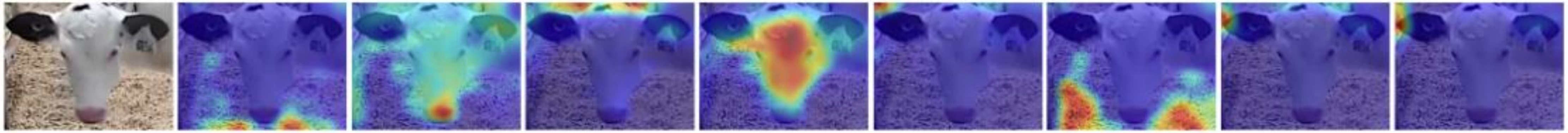
| Video Models (Value in %) | Accuracy | | | | F1-score | | | | Binary Accuracy | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Two Class: | | Three class: | | Two Class: | | Three class: | | Two Class: | | Three class: | |
| | Sample | Whole | Sample | Whole | Sample | Whole | Sample | Whole | Sample | Whole | Sample | Whole |
| Timesformer | 45.59 | **48.53** | 33.82 | 29.41 | **55.42** | 47.76 | 33.82 | 29.41 | **58.89** | 53.67 | 33.53 | 29.18 |
| VideoMAE | 44.12 | **51.47** | 39.71 | 32.35 | 36.67 | **40.00** | 39.71 | 32.35 | 45.02 | **50.58** | 39.39 | 32.08 |

# Model INTR

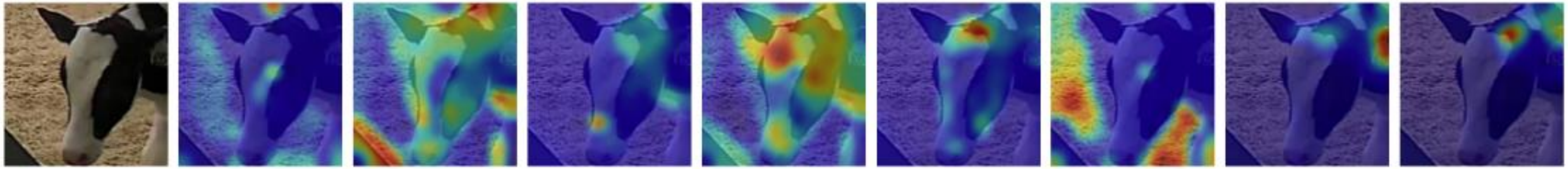
Species predicted by INTR is: Healthy
Species class  is: Healthy



Species predicted by INTR is: Healthy
Species class  is: Healthy

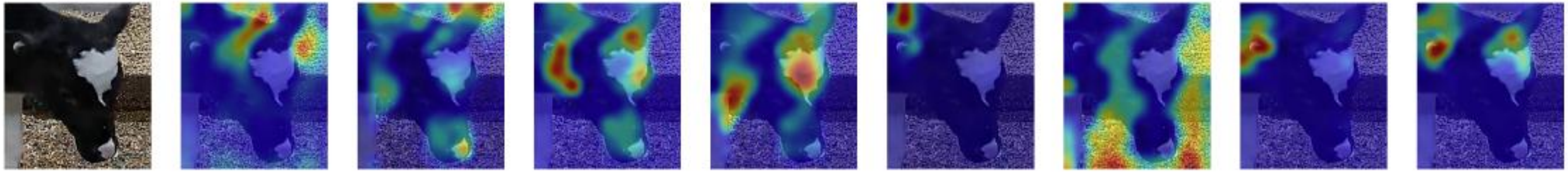


Species predicted by INTR is: Healthy
Species class  is: Illness

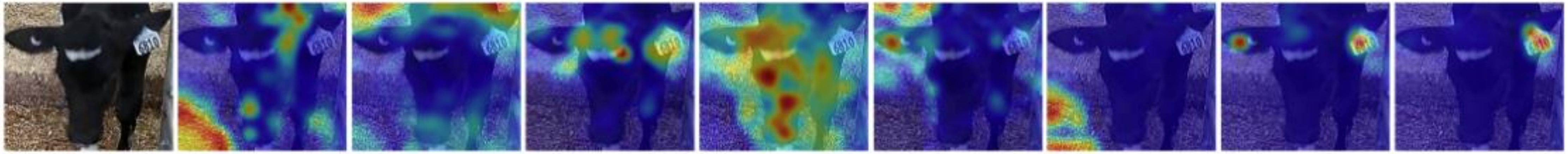


# Model INTR

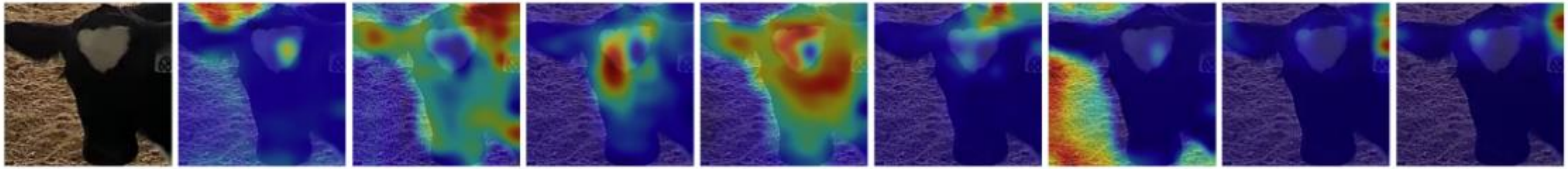
```
Species predicted by INTR is: Illness
Species class  is: Illness
```



```
Species predicted by INTR is: Illness
Species class  is: Illness
```




```
Species predicted by INTR is: Illness
Species class  is: Healthy
```



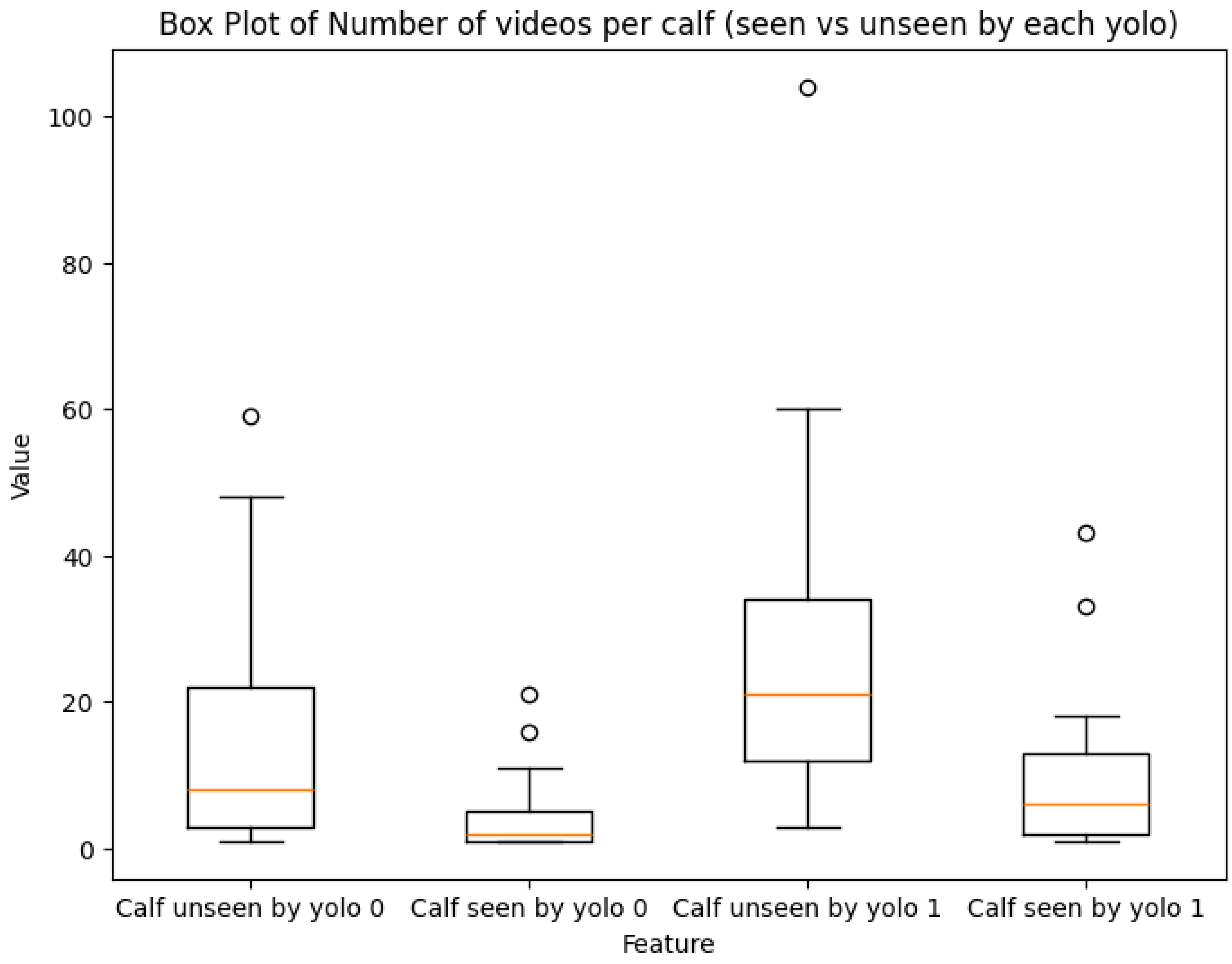
# What next ?

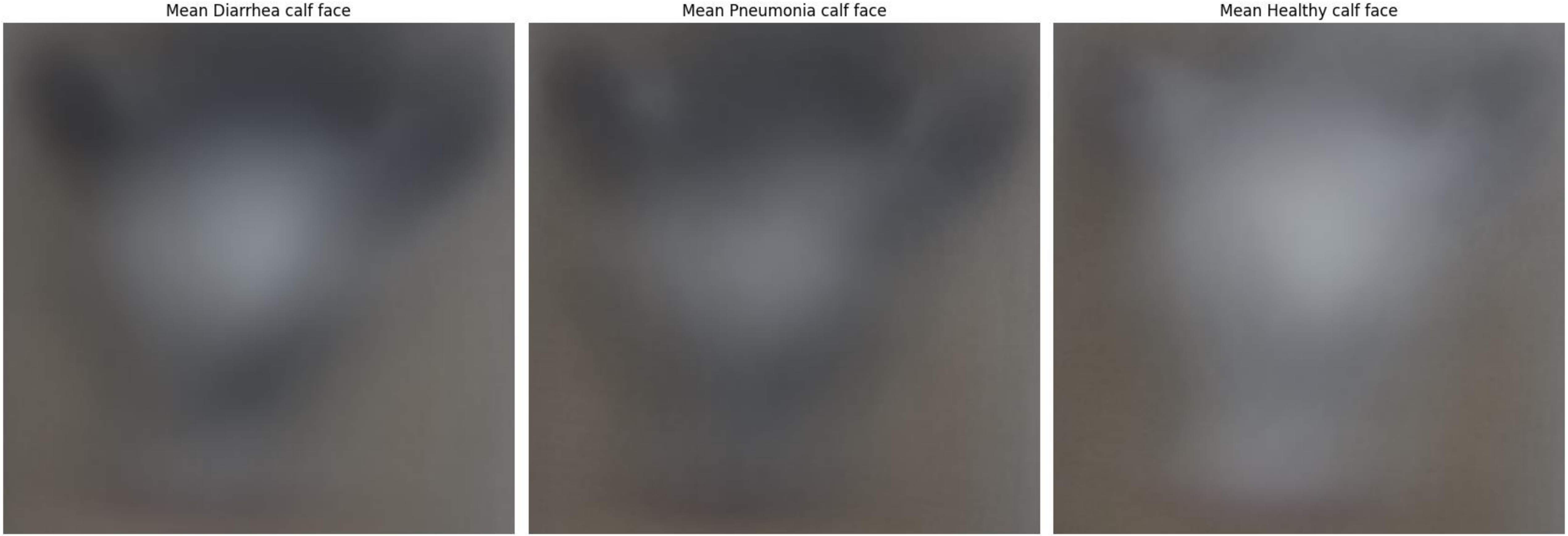
- Find the best hyper-params for each best models
- Test with leaving videos/images if possible
- Test with LSTM+CNN using only calf face as input

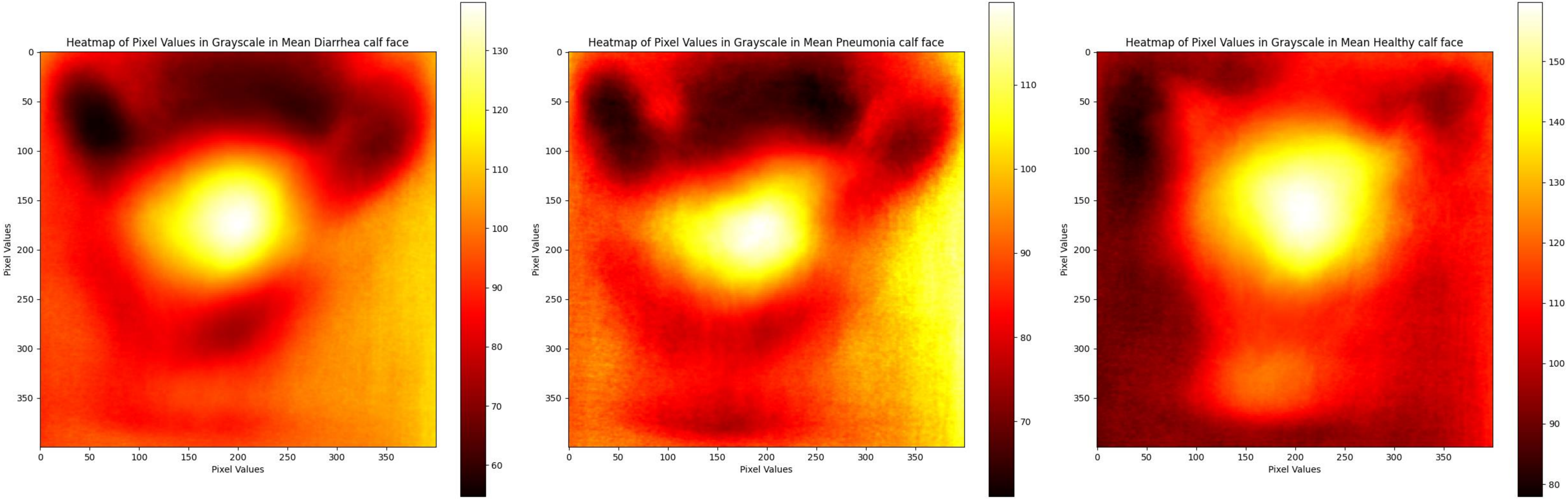
That's all !

# Thanks a lot !

# Notes


Box Plot of Number of videos per calf (seen vs unseen by each yolo)
Value
0
20
40
60
80
100
Calf unseen by yolo 0
Calf seen by yolo 0
Calf unseen by yolo 1
Calf seen by yolo 1
Feature


Mean Diarrhea calf face
Mean Pneumonia calf face
Mean Healthy calf face

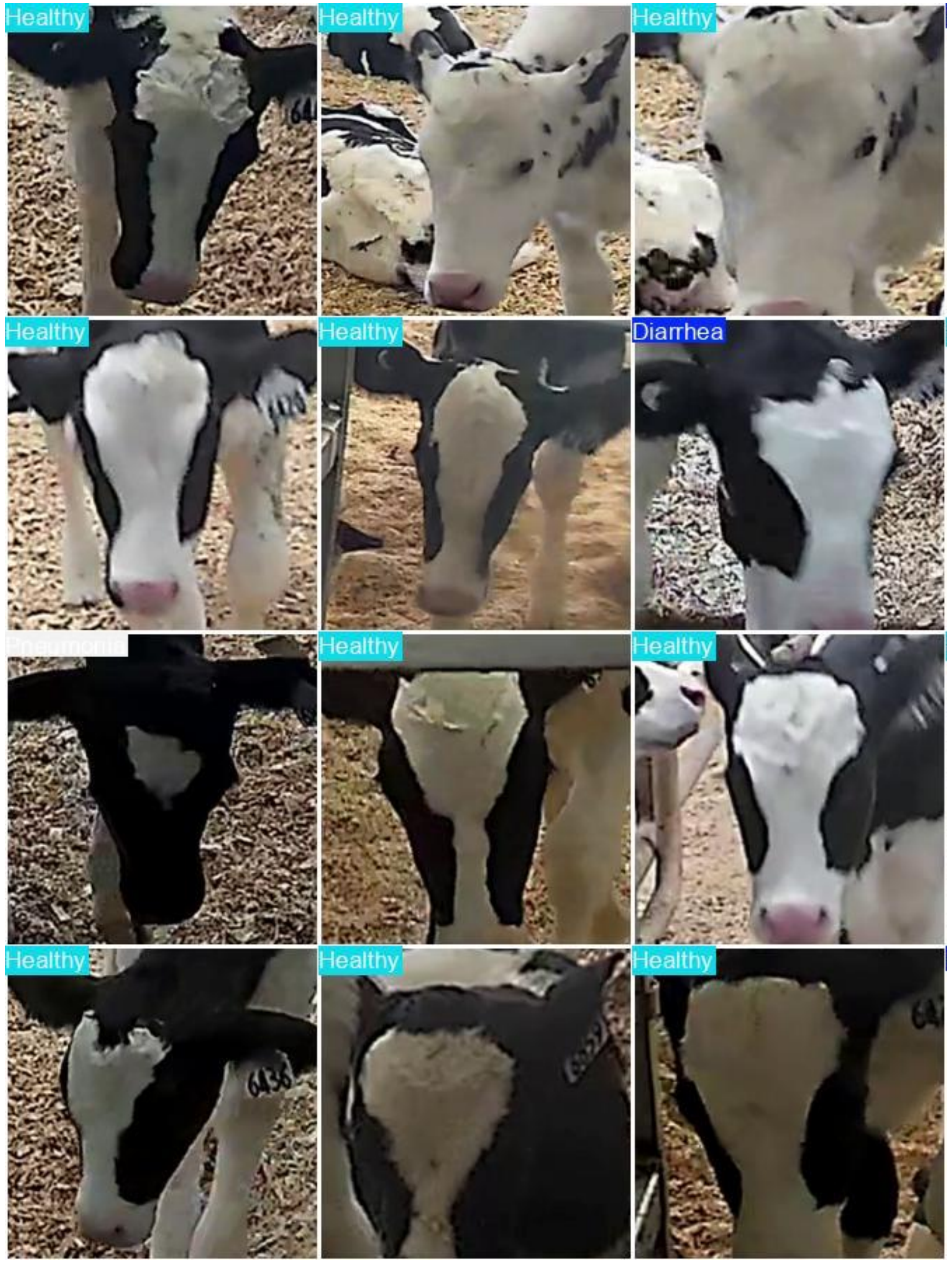


Heatmap of Pixel Values in Grayscale in Mean Diarrhea calf face
Pixel Values
Pixel Values
0 50 100 150 200 250 300 350
60 70 80 90 100 110 120 130
Heatmap of Pixel Values in Grayscale in Mean Pneumonia calf face
Pixel Values
Pixel Values
0 50 100 150 200 250 300 350
70 80 90 100 110
Heatmap of Pixel Values in Grayscale in Mean Healthy calf face
Pixel Values
Pixel Values
0 50 100 150 200 250 300 350
80 90 100 110 120 130 140 150


Image
2022-08-03 16:30:09
HD IPC
Image
2022-08-03 16:30:09
HD IPC

True labels

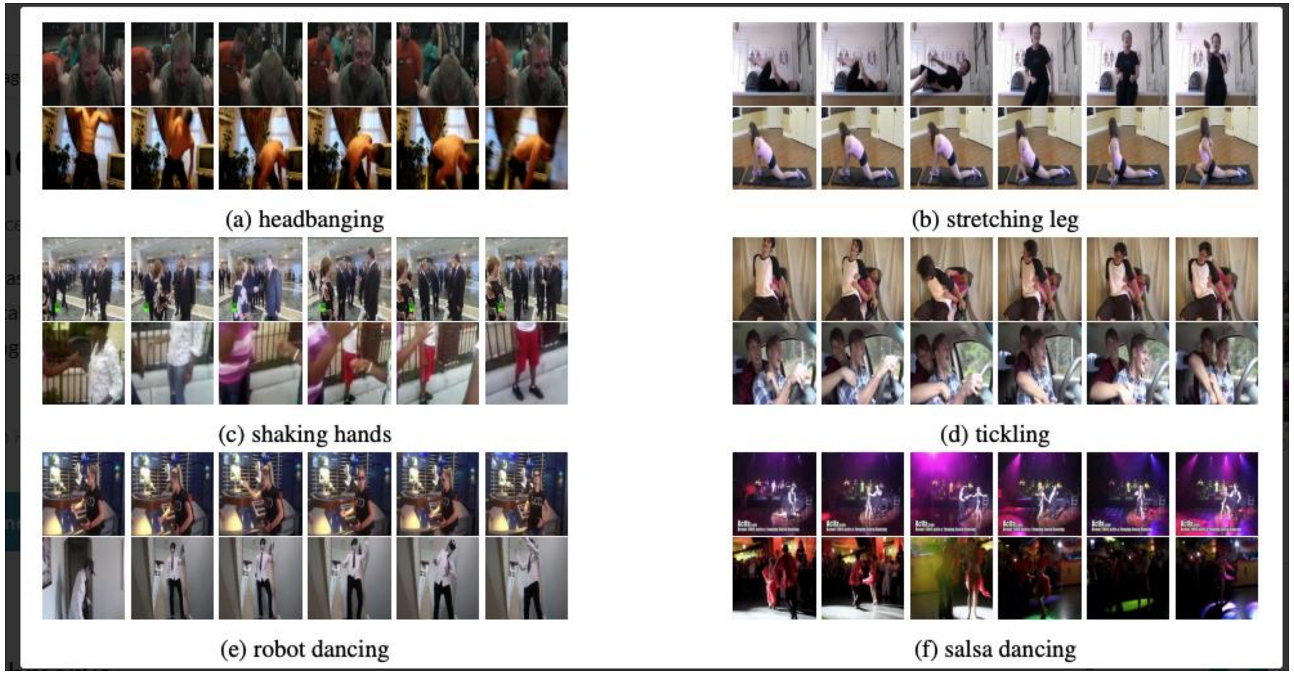

Predicted labels

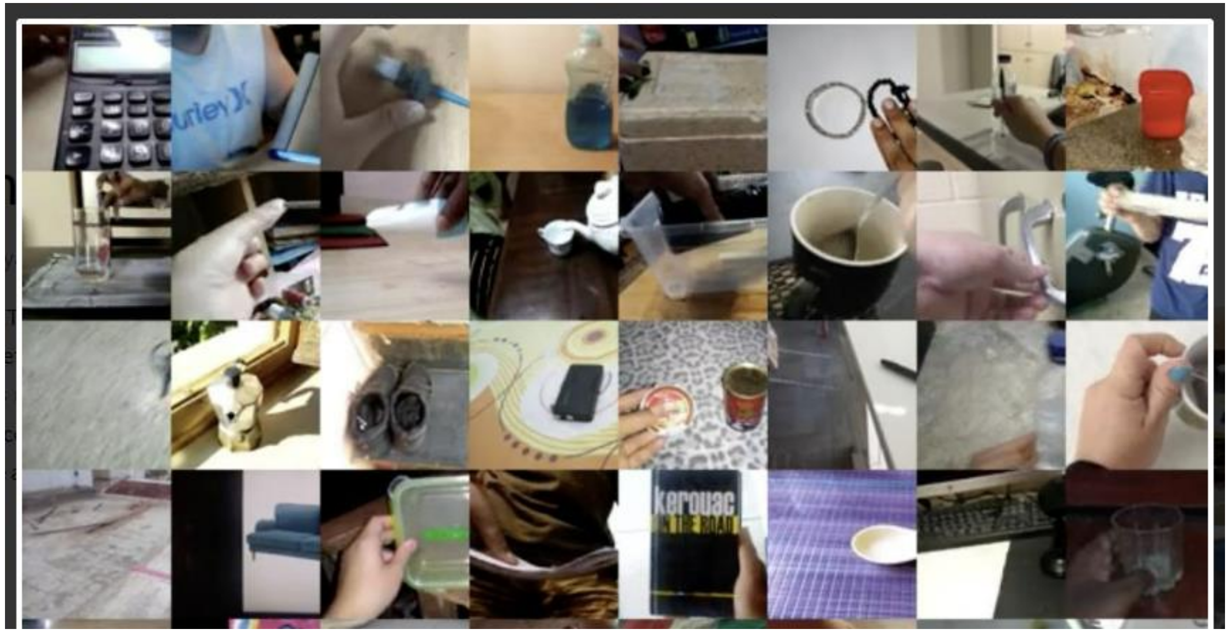


# Modèle VideoMAE

Kinetics dataset

SSv2 dataset



# VideoMAE

- Solution aux redondances entres les frames des vidéos de notre dataset
- Bonnes performances sur de petits datasets

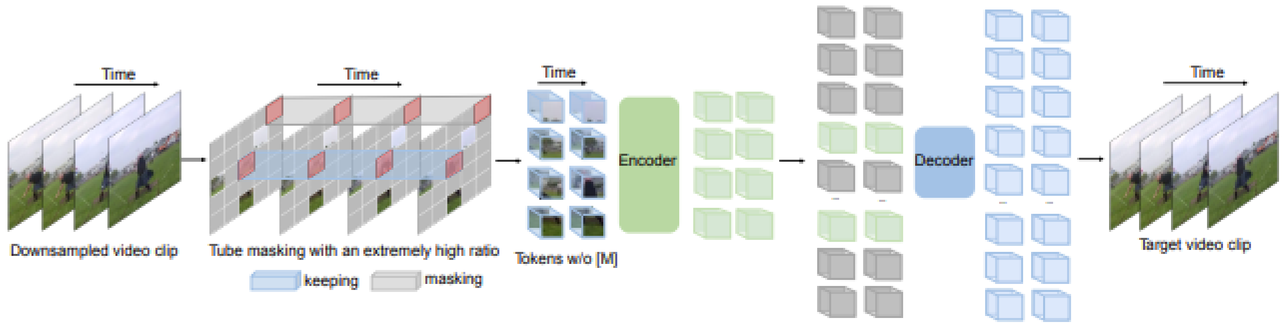



Figure 1: **VideoMAE** performs the task of masking random cubes and reconstructing the missing ones with an asymmetric encoder-decoder architecture. Due to high redundancy and temporal correlation in videos, we present the customized design of tube masking with an extremely high ratio (90% to 95%). This simple design enables us to create a more challenging and meaningful self-supervised task to make the learned representations capture more useful spatiotemporal structures.


Figure 2: Slowness is a general prior in (a) video data [88]. This leads to two important characteristics in time: temporal redundancy and temporal correlation. Temporal redundancy makes it possible to recover pixels under an extremely high masking ratio. Temporal correlation leads to easily reconstruct the missing pixels by finding those corresponding patches in adjacent frames under plain (b) frame masking or (c) random masking. To avoid this simple task and encourage learning representative representation, we propose a (d) tube masking, where the masking map is the same for all frames.

*Figure 7.* Visualization of space-time attention from the output token to the input space on Something-Something-V2. Our model learns to focus on the relevant parts in the video in order to perform spatiotemporal reasoning.

Timesformer

- Faster to train than 3D CNN
- higher test efficiency (at a small drop in accuracy)

# ViT

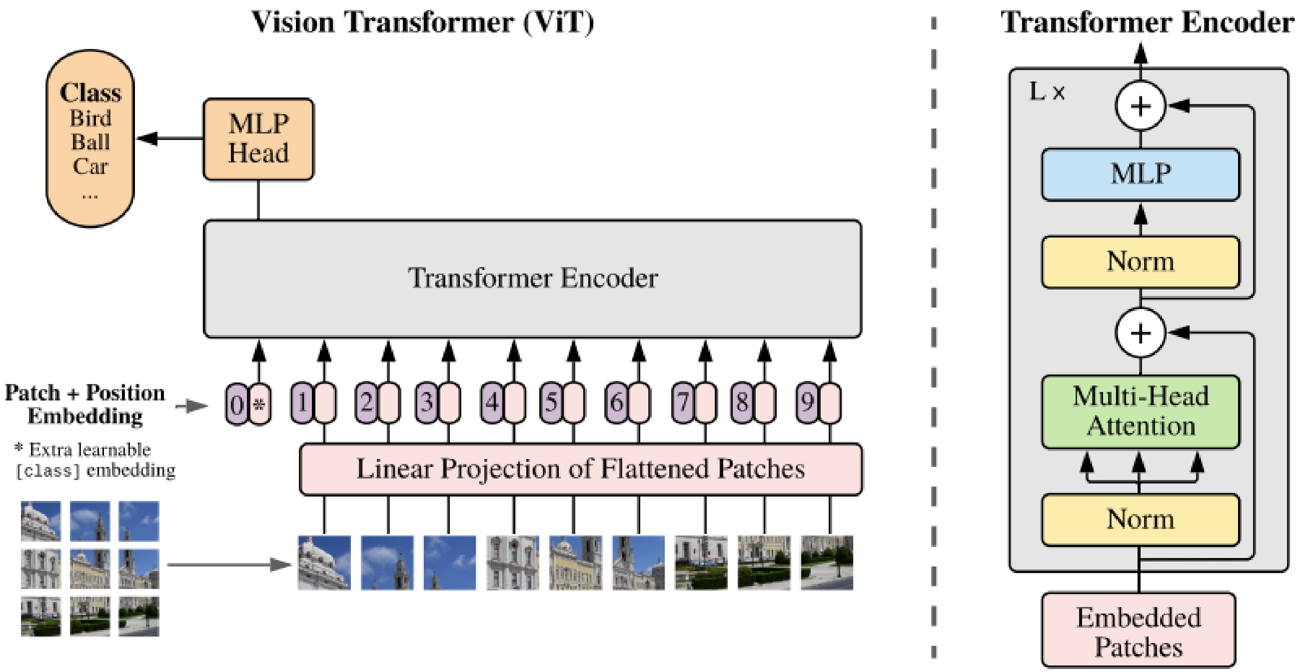



Figure 1: Model overview. We split an image into fixed-size patches, linearly embed each of them, add position embeddings, and feed the resulting sequence of vectors to a standard Transformer encoder. In order to perform classification, we use the standard approach of adding an extra learnable "classification token" to the sequence. The illustration of the Transformer encoder was inspired by Vaswani et al. (2017).

# Efficientnet-b3

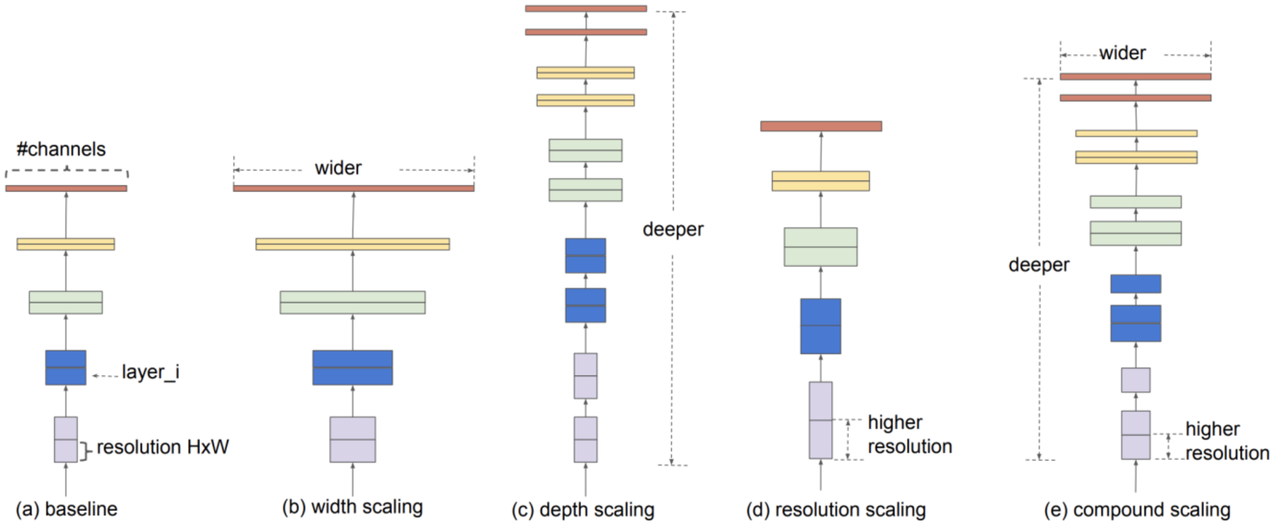
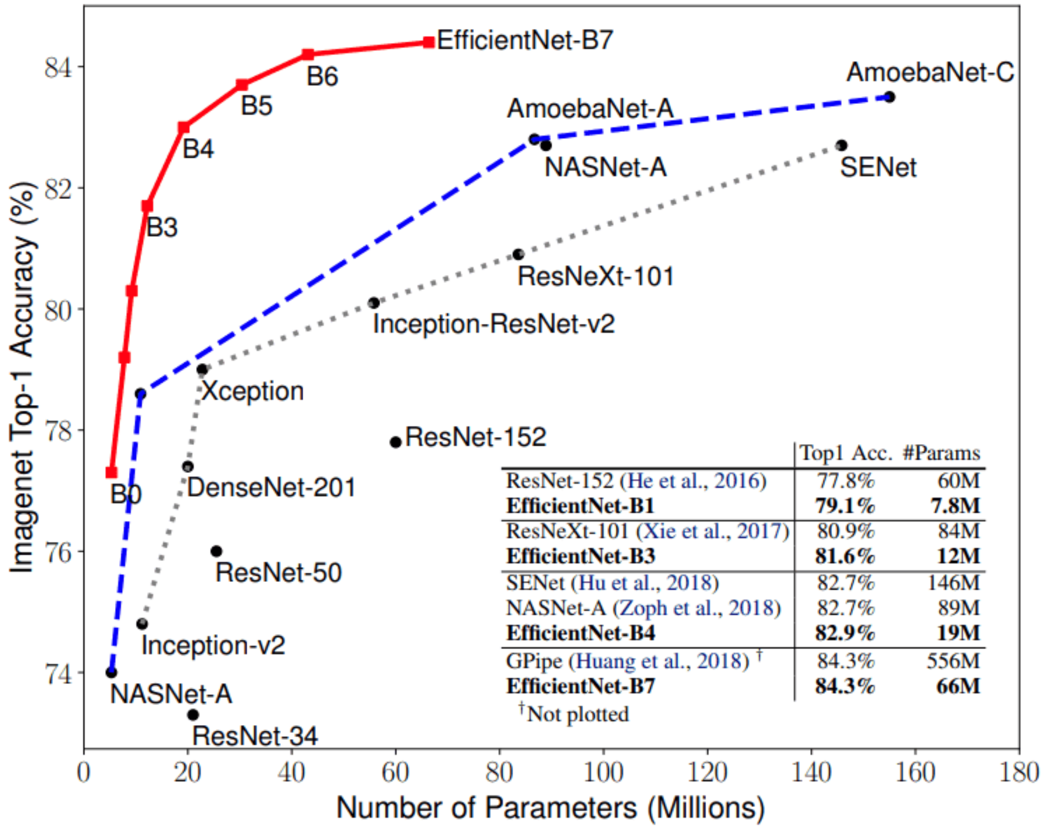


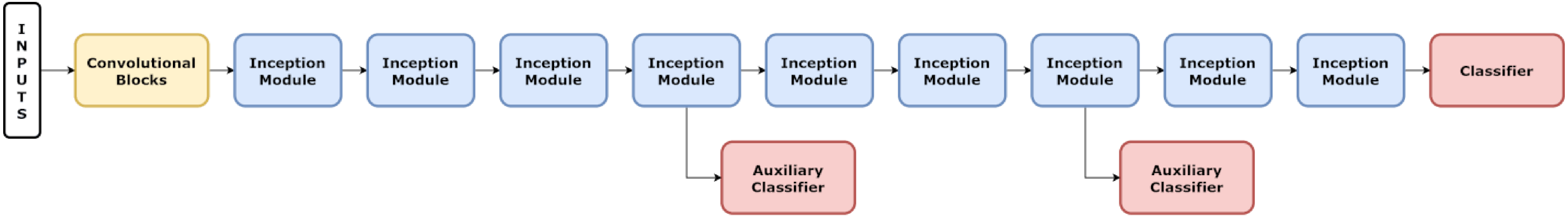

*Figure 2.* **Model Scaling.** (a) is a baseline network example; (b)-(d) are conventional scaling that only increases one dimension of network width, depth, or resolution. (e) is our proposed compound scaling method that uniformly scales all three dimensions with a fixed ratio.




| | Top1 Acc. | #Params |
|---|---|---|
| ResNet-152 (He et al., 2016) | 77.8% | 60M |
| **EfficientNet-B1** | **79.1%** | **7.8M** |
| ResNeXt-101 (Xie et al., 2017) | 80.9% | 84M |
| **EfficientNet-B3** | **81.6%** | **12M** |
| SENet (Hu et al., 2018) | 82.7% | 146M |
| NASNet-A (Zoph et al., 2018) | 82.7% | 89M |
| **EfficientNet-B4** | **82.9%** | **19M** |
| GPipe (Huang et al., 2018) † | 84.3% | 556M |
| **EfficientNet-B7** | **84.3%** | **66M** |

†Not plotted

# InceptionV3



# INterpretable TRansformer

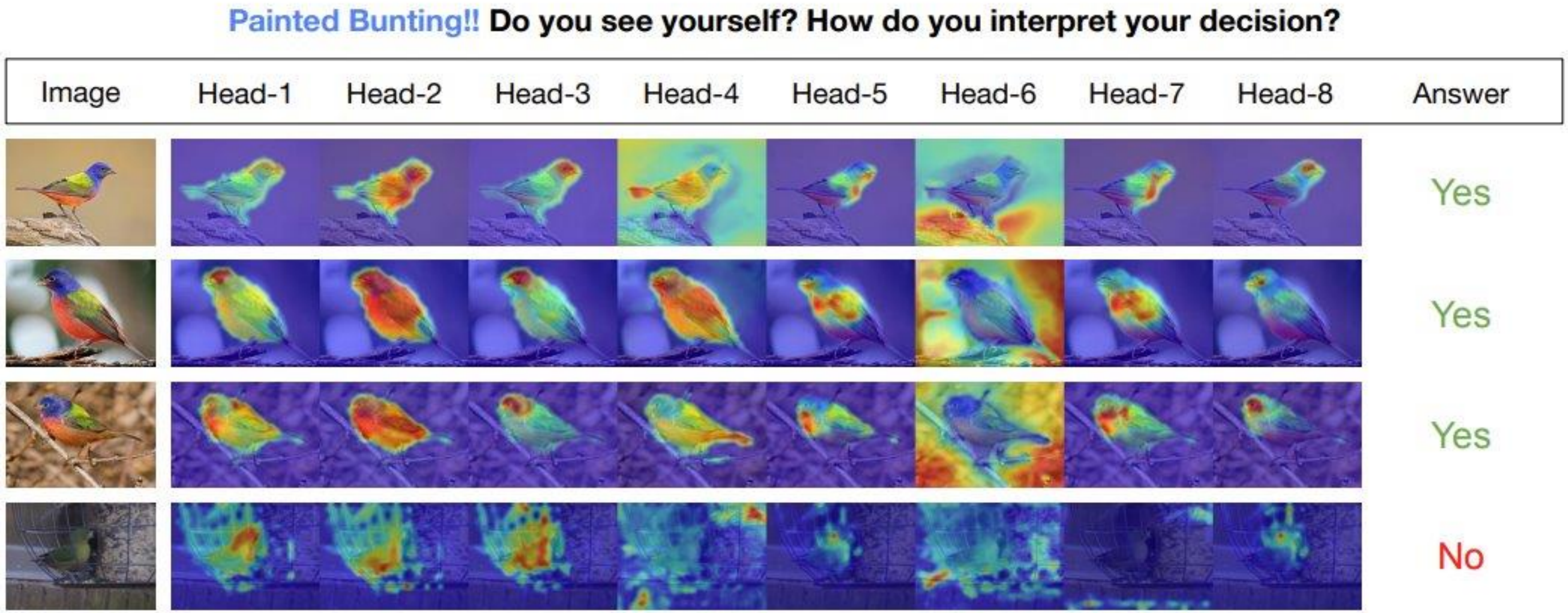



Figure 1: **Illustration of INTR**. We show four images (row-wise) of the same bird species Painted Bunting and the eight-head cross-attention maps (column-wise) triggered by the query of the ground-truth class. Each head is learned to attend to a different (across columns) but consistent (across rows) semantic cue in the image that is useful to recognize this bird species (e.g., attributes). The exception is the last row, which shows inconsistent attention. Indeed, this is a misclassified case, showcasing how INTR interprets (wrong) predictions.