# ASD project

## Introduction

**Problem**:

To Detect and/or Predict emotional meltdowns of autistic signals based on simple signals that can be recovered by portable devices (e.g. mobile phone, smartwatch).

**Use case**:

Children and families of children with autism and other developmental disorders can use this app to detect meltdowns and handle them accordingly.

## Requirements

1. Fast response (fast execution of the detection algorithm)
2. Good accuracy in terms of sensitivity, ideally all meltdowns should be detected. Specificity and false positives are not necessarily an issue, since an unnecessary soothing process is not detrimental to the child.
3. The app should conform with private information protection and GDPR regulations.
4. Decision making based on multiple and varied signals including image, movement, and heartrate data.

## Approach

Modern machine learning and artificial intelligence algorithms provide various solutions for image manipulation. Common technologies for the task of emotion detection include deep neural networks such as VGG and EfficientNet. These networks can be pretrained on standard image classification tasks and then optimized to be able to perform classification with custom classes.

And while that's a good skill to have, our models haven't been performing as well as we'd like.

That's where **transfer learning** comes in.

The whole idea of transfer learning is to **take an already well-performing model on a problem-space similar to yours and then customising it to your use case**.

Since we're working on a computer vision problem (image classification with FoodVision Mini), we can find pretrained classification models in [torchvision.models](https://pytorch.org/vision/stable/models.html#classification).

|  |  |
| --- | --- |
| [ResNet](https://arxiv.org/abs/1512.03385)'s | torchvision.models.resnet18(), torchvision.models.resnet50()... |
| [VGG](https://arxiv.org/abs/1409.1556) (similar to what we used for TinyVGG) | torchvision.models.vgg16() |
| [EfficientNet](https://arxiv.org/abs/1905.11946)'s | torchvision.models.efficientnet\_b0(), torchvision.models.efficientnet\_b1()... |
| [VisionTransformer](https://arxiv.org/abs/2010.11929) (ViT's) | torchvision.models.vit\_b\_16(), torchvision.models.vit\_b\_32()... |
| [ConvNeXt](https://arxiv.org/abs/2201.03545) | torchvision.models.convnext\_tiny(), torchvision.models.convnext\_small()... |
| More available in torchvision.models |  |

ou might think better performance is *always better*, right?

That's true but **some better performing models are too big for some devices**.

For example, say you'd like to run your model on a mobile-device, you'll have to take into account the limited compute resources on the device, thus you'd be looking for a smaller model.

Understanding this **performance vs. speed vs. size tradeoff** will come with time and practice.

For me, I've found a nice balance in the efficientnet\_bX models.

## Dataset

An image database of faces of children, classified as autistic and non-autistic. The images were downloaded manually from images available on the internet and cross-references with article content and descriptions. All images are center cropped around the child’s face. The images are of varied height and width but were resized to 225x225 pixels for further experimentation.

Table - Breakdown of the Image Dataset

|  |  |
| --- | --- |
| Classification | Number of images |
| Autistic | 1463 |
| Non-Autistic | 1463 |
| Total | 2926 |

Note:

* There is a possibility that there is are some cases of autism to be present in the non-autistic dataset, as it is not easy to discard the possibility as a result of the way the dataset was created, i.e. there isn’t a way to search for non-autistic children, one can only look up images of children.
* Image resizing introduces noise and may affect performance of the models.

Table - Example of images included in the dataset

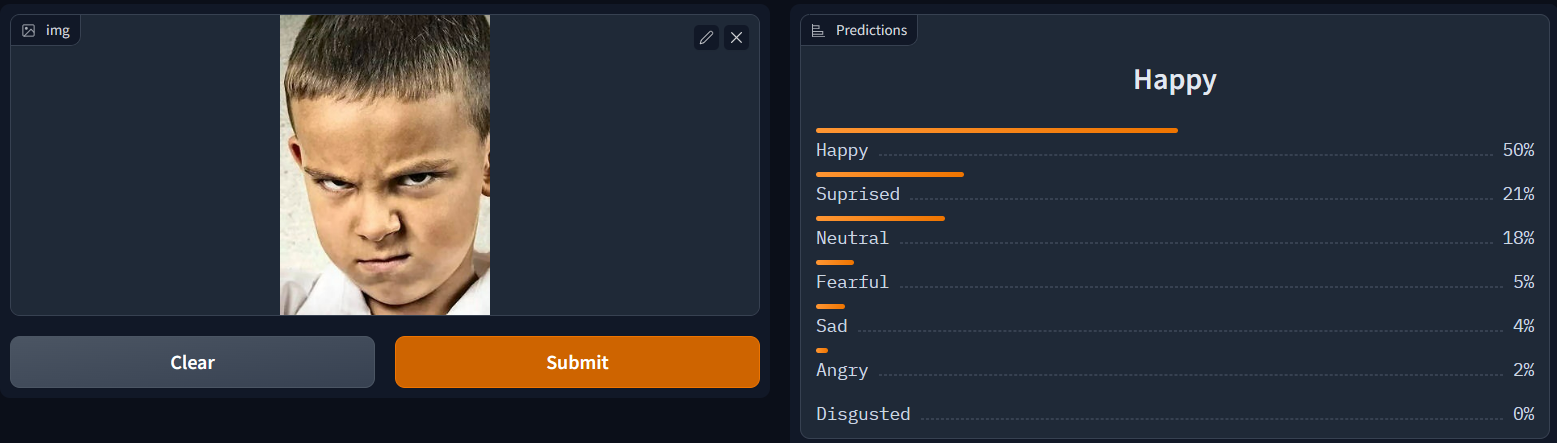
|  |  |  |  |
| --- | --- | --- | --- |
| Autistic |  |  |  |
| **Non-autistic** |  |  |  |

## System architecture

Modern machine learning and artificial intelligence algorithms provide various solutions for image manipulation.

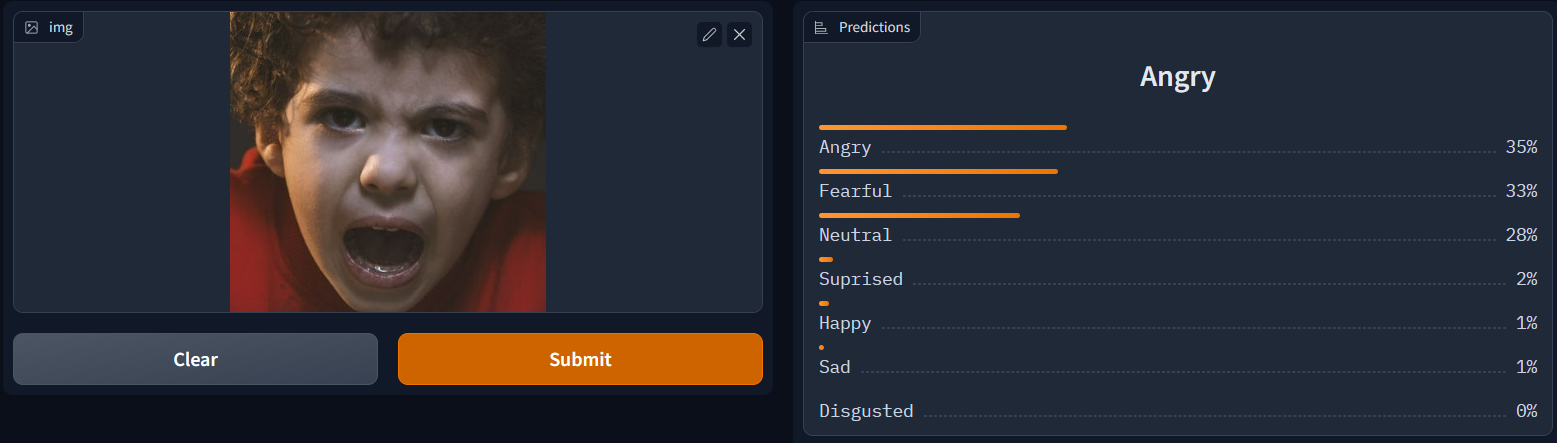
## Experiments

### With EfficientNet



A screenshot of a computer

Description automatically generated with medium confidence



For the total of images of autistic children, most images were classified as ‘Happy’ or ‘Neutral’. Those two classes also had the highest probability for the predicted emotion, at 71% and 60% respectively.

Graphical user interface, text, application

Description automatically generated

### With Transformers

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

Text, application

Description automatically generated

## Optimization and Future Directions

Erer

## Links

erer