

Two Headed Classifier Use Case

Argo Saakyan

Computer Vision Engineer

Idea & Constrains

Classification? - Easy!

But what about constrains?

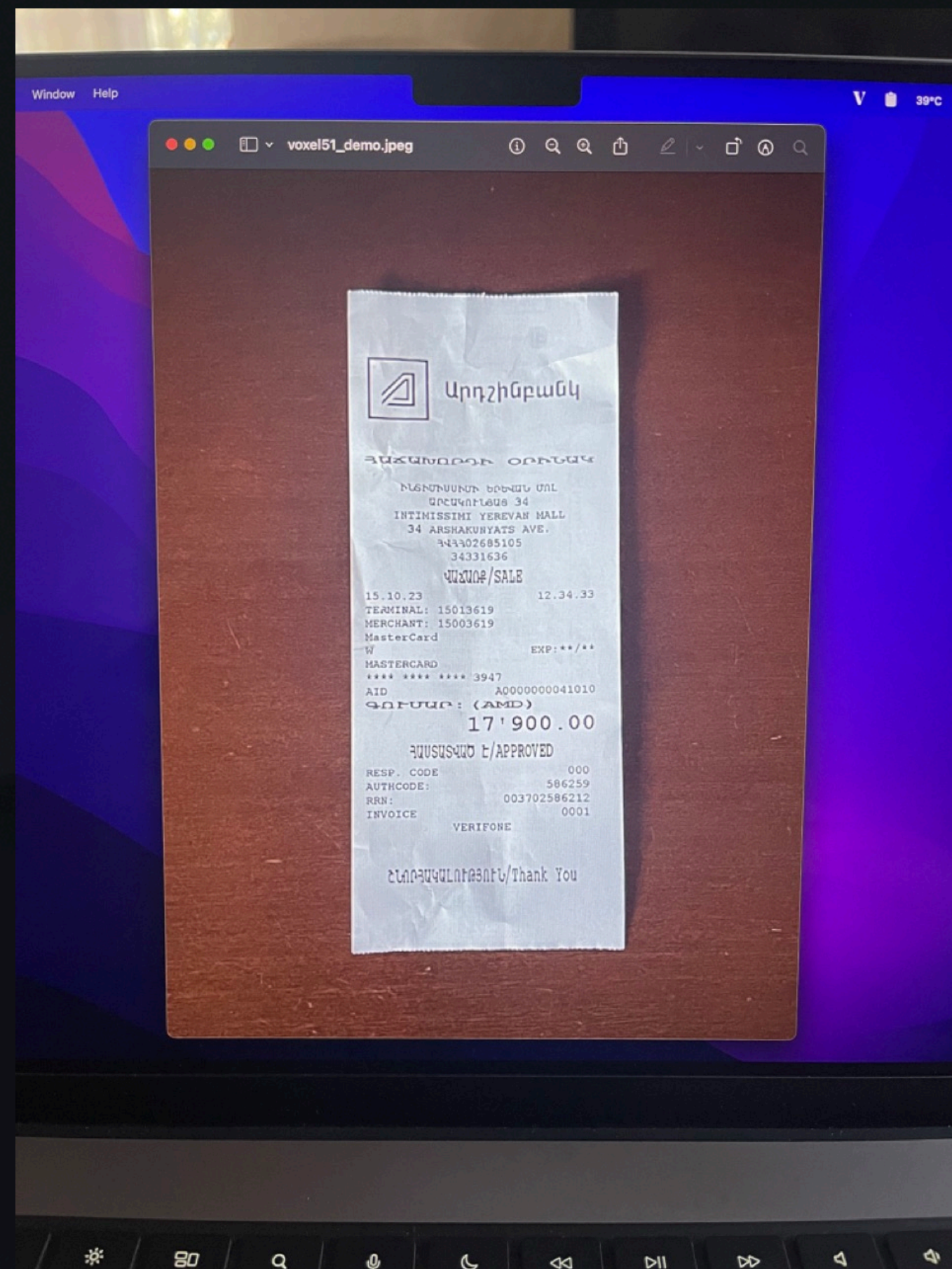
- Size of the model;
- Inference speed;
- Exportability;
- Amount of models;



The Task



Real document



Screen



Not a document

Dataset

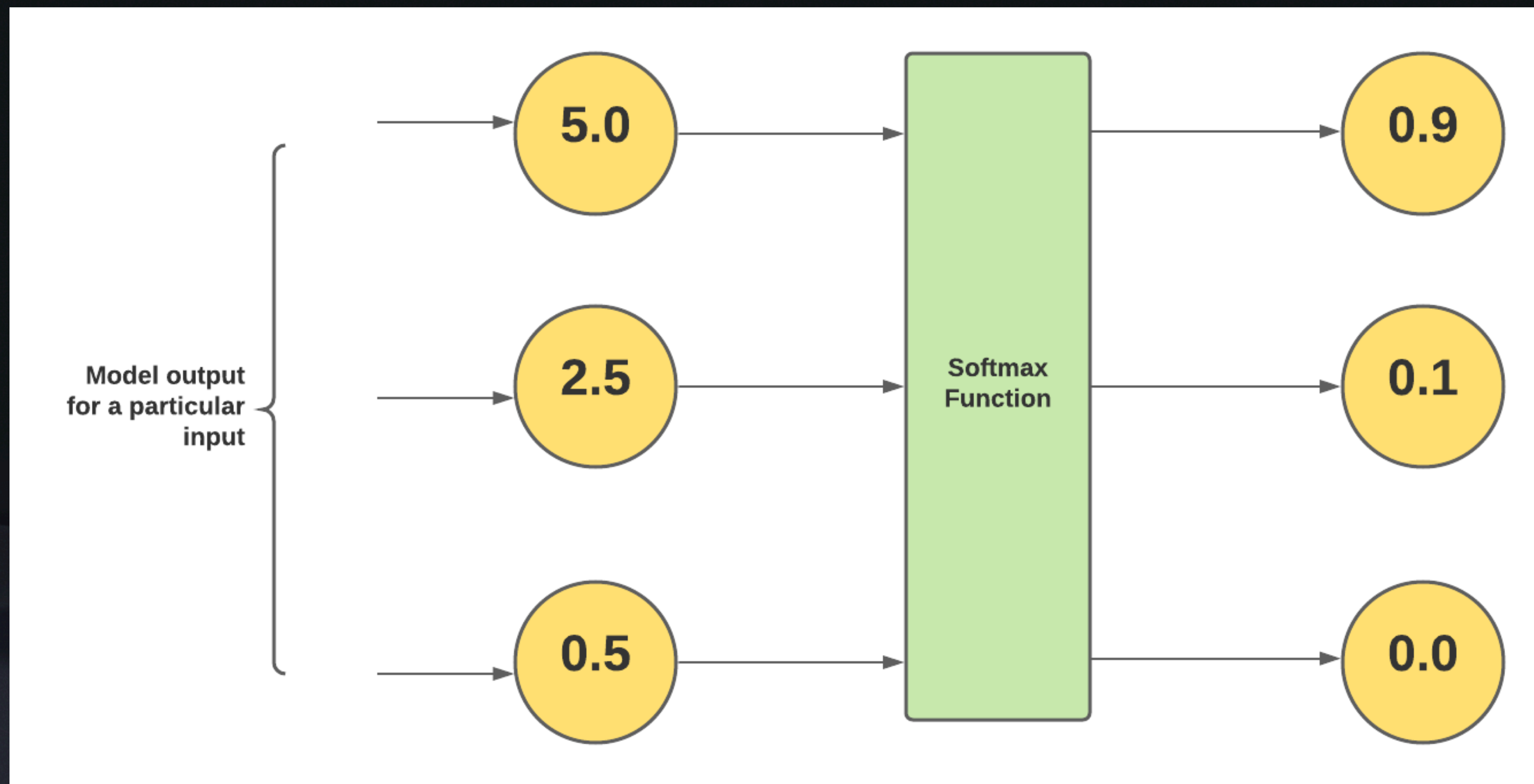
Structure

```
dataset/  
├── documents/  
│   ├── img_1.jpg  
│   ├── ...  
│   └── img_100.jpg  
├── screens/  
│   ├── img_1.jpg  
│   ├── ...  
│   └── img_100.jpg  
├── not a documents/  
│   ├── img_1.jpg  
│   ├── ...  
│   └── img_100.jpg  
├── train.csv  
├── val.csv  
└── test.csv
```

Csv files

```
documents/img_1.jpg      | 0  
not a document/img_1.jpg | 1  
screens/img_1.jpg       | 2  
...
```

Three output neurons



Two headed architecture

```
def _forward_impl(self, x):  
    # See note [TorchScript super()]  
    x = self.conv1(x)  
    x = self.maxpool(x)  
    x = self.stage2(x)  
    x = self.stage3(x)  
    x = self.stage4(x)  
    x = self.conv5(x)  
    x = x.mean([2, 3]) # globalpool  
    x = self.fc(x)  
    return x
```

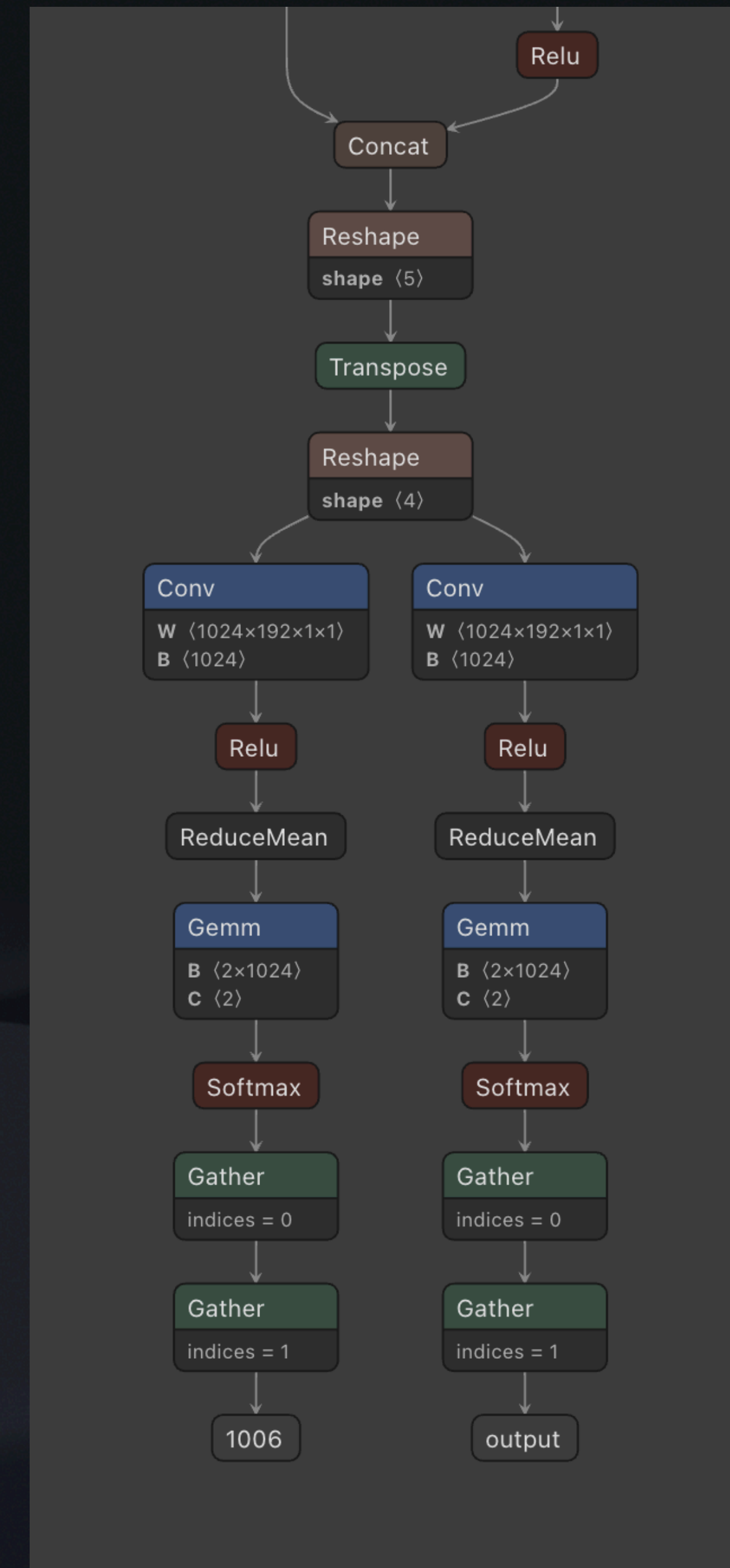
```
def forward(self, x: torch.Tensor) -> torch.Tensor:  
    x = self.base_model.conv1(x)  
    x = self.base_model.maxpool(x)  
    x = self.base_model.stage2(x)  
    x = self.base_model.stage3(x)  
    x = self.base_model.stage4(x)  
  
    # Pass through the separate convolutions for each head  
    x1 = self.head1_conv(x)  
    x1 = x1.mean([2, 3]) # globalpool for first head  
    out1 = self.fc1(x1)  
  
    x2 = self.head2_conv(x)  
    x2 = x2.mean([2, 3]) # globalpool for second head  
    out2 = self.fc2(x2)  
    return out1, out2
```

Pros:

- 1 model for 2 tasks
- Good accuracy
- More control

Cons:

- 6% speed loss
- 40% size increase



Let's see some code...

Results

Model (img size)	Precision	Recall	Latency (s)*
Three output neurons (256)	0.993	0.855	0.027
Three output neurons (320)	1.0	0.846	0.029
Two heads (256)	1.0	0.873	0.029

Latency (s)* - mean inference time on 1 image, including transforms and softmax.

To sum up...

- Classification is easy, but it gets harder with all real world constrains
- Optimize subtasks and try not to create K models for every big task
- Customize models and training pipelines to have a better control
- Test your hypothesis, run experiments and save results (hydra, wandb...)

Thanks for your attention!

- LinkedIn - [linkedin.com/in/argo-saakyan](https://www.linkedin.com/in/argo-saakyan)
- Repo - https://github.com/ArgoHA/two_headed_classifier
- Article - <https://medium.com/p/c8dc4f684091>

