



# **Decentralized CNN prediction service on top of Golem**

# Issues

- Training deep convolutional nets is a lengthy process, usually days or even weeks on powerful dedicated systems
- Training deep convolutional nets on Golem is thus very difficult, if not impossible
- This difficulty to maintain status over long periods of time is a known feature of decentralized systems, it's a beauty though
- Individuals in decentralized communities pop in for short periods of time and then leave
- This is an issue when long maintenance is required



# Solution

- Train popular models on degen centralized systems and transfer weights of a ready model to Golem nodes
- Centralized systems are costly when scalability is required(e.g. imagine having rented 20 AWS servers to serve basic prediction)
- A cheap solutions is using Golem nodes on a pay per request model

# Rational

- Frame popular CNN models as Golem provider images
- The ideal is having an image per model, this would boost scalability
- Some models are small, so to save space we pack them together
- Put a requestor agent behind a REST API to serve various image classification requests
- The request-response cycle is indeed **asynchronous**
- Batch and schedule requests to boost earnings and scalability

# Batching & scheduling

- Group requests based on their requested models
- Divide requests into chunks of 16 requests per task
- Chunks of 16 puts a 64mb payload max cap on any task
- On a regular basis, call a script to send requests to nodes for prediction
- Collect results and pt them into a db
- Prediction results are ready!



# Available models

- DenseNet121
- DenseNet169
- DenseNet201
- InceptionResNetV2
- InceptionV3
- Xception
- The rest to be added...



# How to request a prediction or query the result?

- To request a prediction:

```
curl -X POST -F image=@kitten.jpg 'http://157.90.197.235/requestPrediction/[model]'
```

- To query the results:

```
curl -X POST 'http://157.90.197.235/readPrediction/[request-id]'
```



**That's it, TY!**

<https://github.com/rezahsnz/mlg>