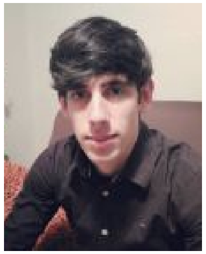
Seminar Project

Team Presentation

Team Composition

Albert Jiménez
Telecom engineer
MsC

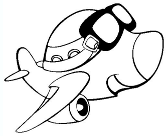


Marc Górriz Blanch
Telecom engineer
BsC

Dennj Osele
Automation engineer

Michele Compri
Telecom engineer
MsC

Adria Romero
Telecom engineer
BsC



Task 1

Architecture

Task 1

Hardware

- Server
  - **Very Slow**: 6~7h to train MNIST with 12 epochs

- Our own computers
  - Faster than Server but still **slow**
  - 1~2h to train MNIST with 12 epochs
  - Only 2 available

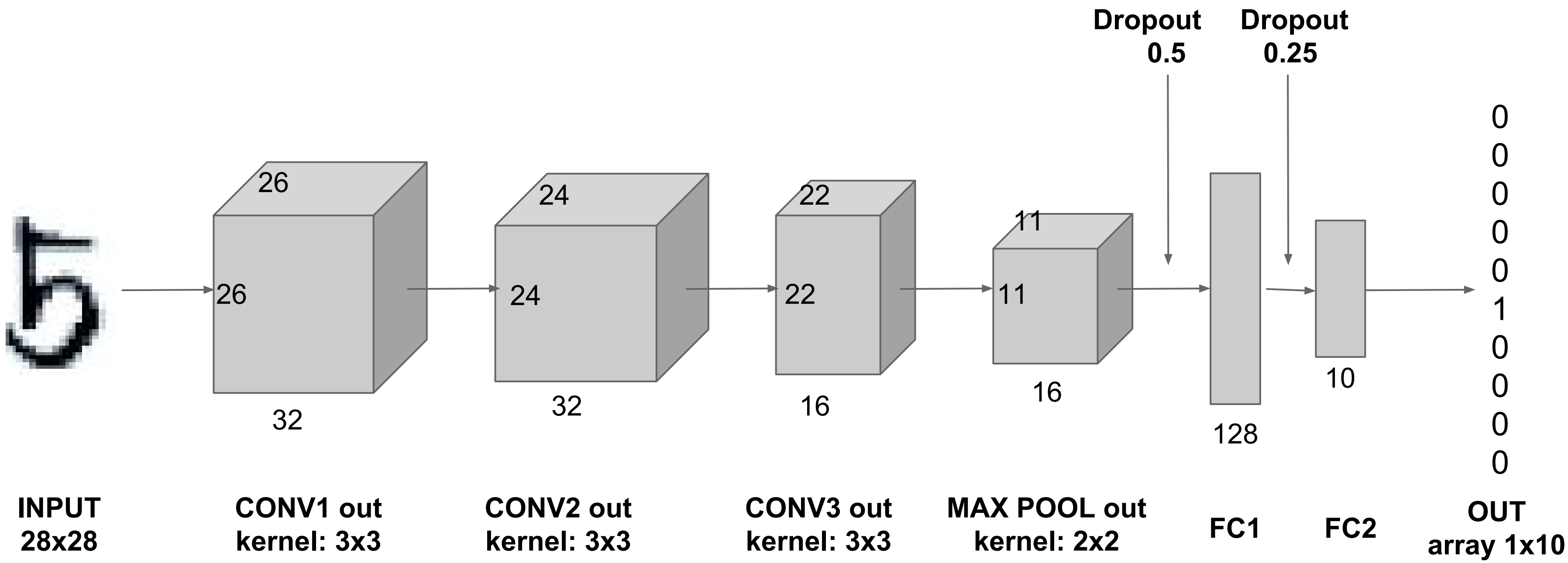
Problem: Computational Bottleneck !

Task 1

Software

- Keras
  - Python language
  - **Simplicity**

Custom Architecture



- We use it to train MNIST dataset
- Adadelta optimizer
- Batches of 128 images
- 1h to train

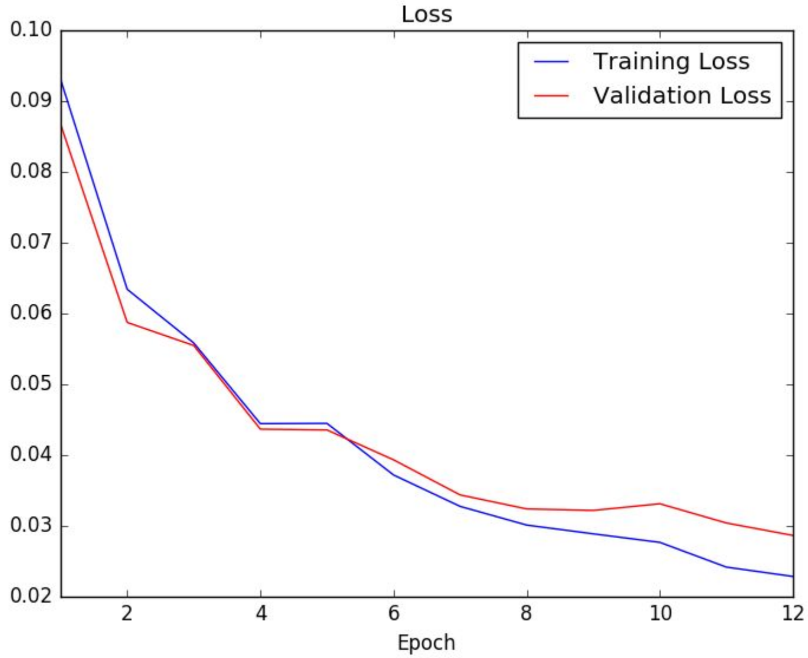
Parameters Table

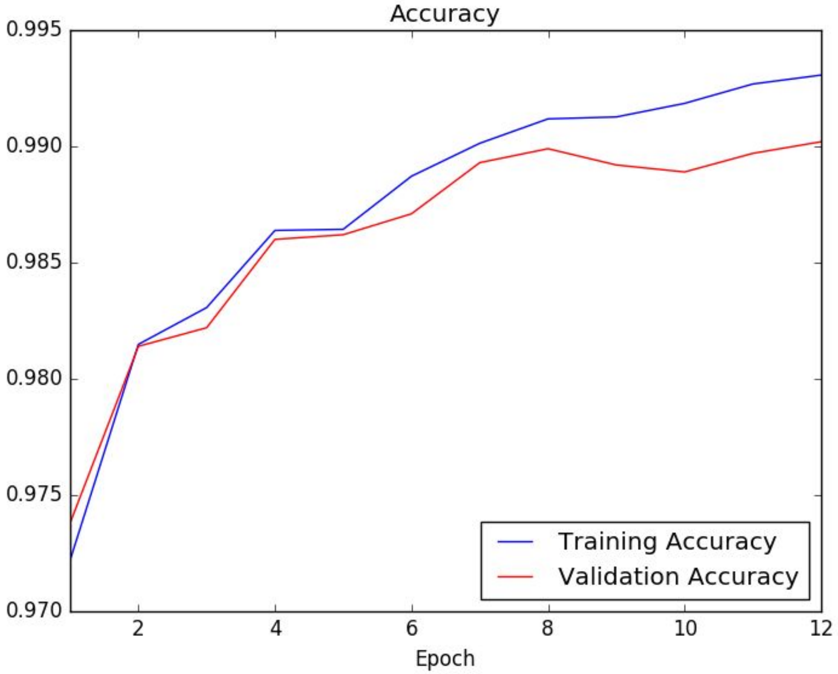
	Parameters	Memory
Conv 1	**320**	**~ 1.25 kB**
Conv 2	**9248**	**~ 36.125 kB**
Conv 3	**4624**	**~18 kB**
FC 1	**247936**	**~ 968.5 kB**
FC 2	**1290**	**~ 5 kB**
Total	**263418**	**~ 1MB**

Task 2

Results

MNIST





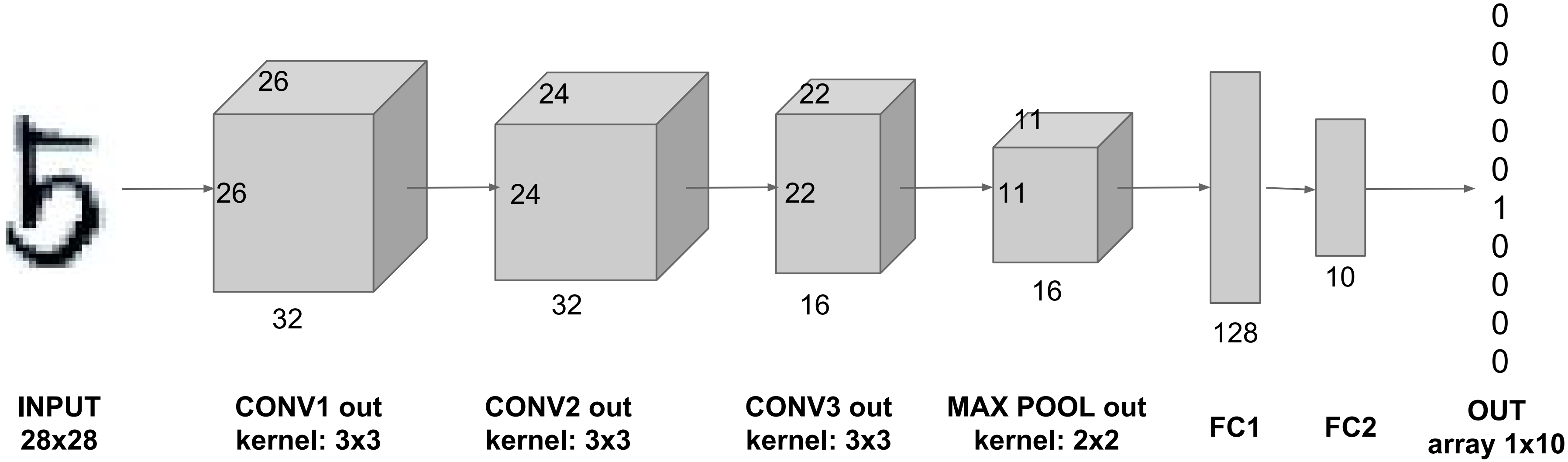
We achieve a 99% accuracy

Task 2

Training

Task 2

Proposed Architecture



- We use it to train MNIST dataset
- Adadelta optimizer
- Batches of 128 images

Overfitting

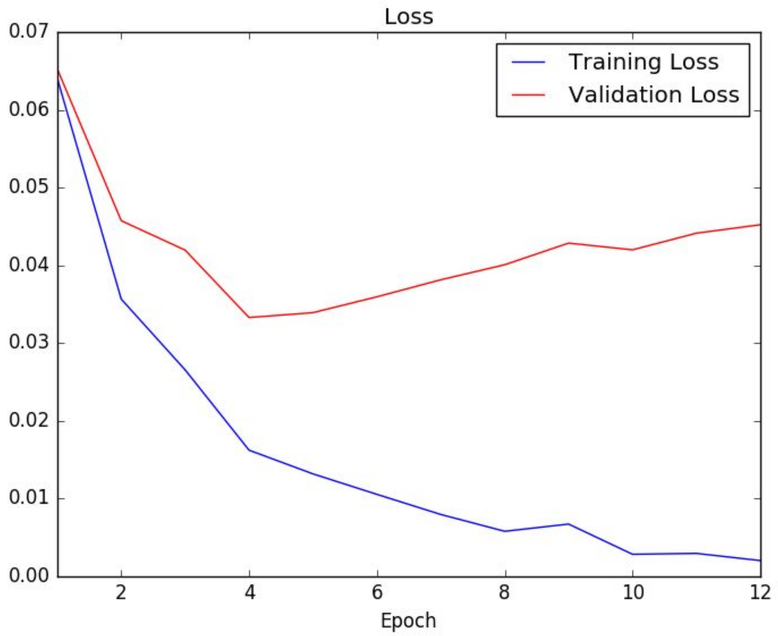
- We create overfitting by removing the dropout layers and increasing the number of parameters of our FC layer

- We observe the difference with the proposed model

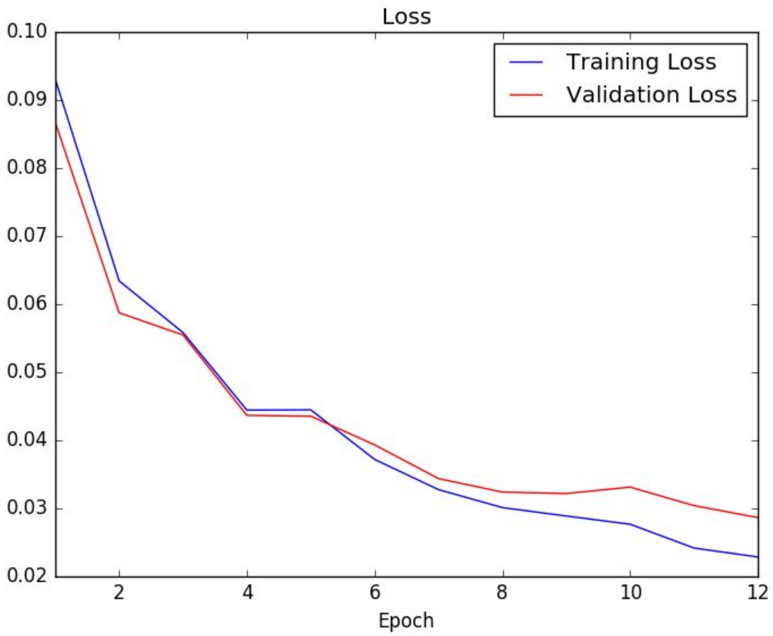
Task 2

Training

Overfitting and solutions in MNIST



Without dropout and adding complexity (More parameters on FC)



Our proposed Architecture

Changing the batch size

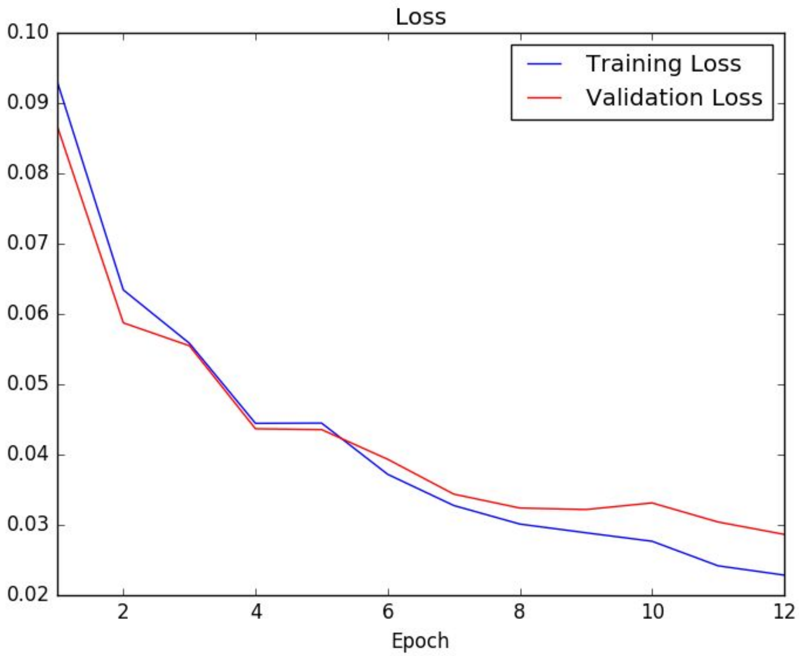
- Larger batches = More data available when uploading the weights = Better update

- We observe the difference with the proposed model

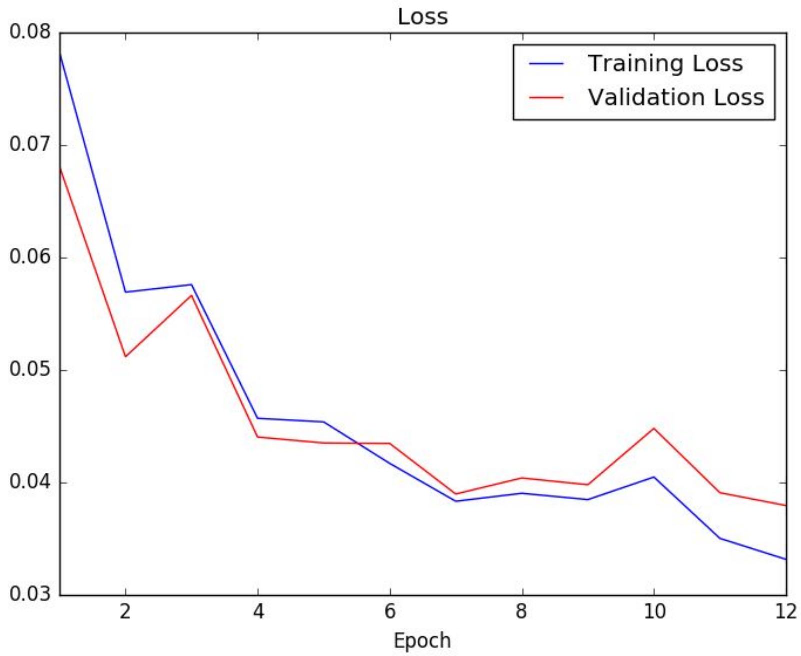
Task 2

Training

Change the size of batches in MNIST dataset.



128 images



8 images

Batch Normalization

- Normalize using the statistics from the batches
- Improves convergence speed
- Can act as a regularizer
- A bit slower to train

Task 2

Batch Normalization

Batch normalization on CIFAR-10

Without norm **Batch normalized**

Data Augmentation

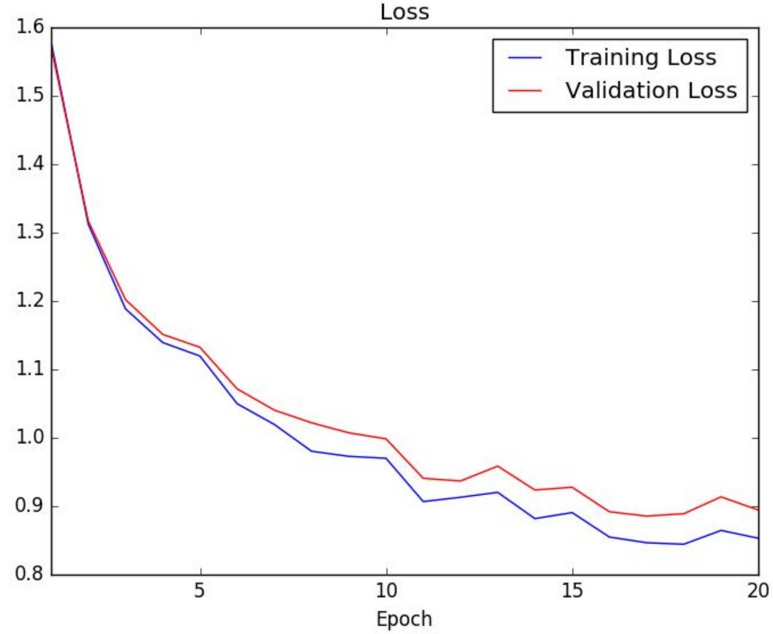
- Feed the network with modified data to increase its invariance to rotation, scaling, translation…

- In our experiment maybe adding noise. We would need more epochs to perform a fair comparison.

Training

Real time Data augmentation in CIFAR10

Without data augmentation



With data augmentation

Task 3

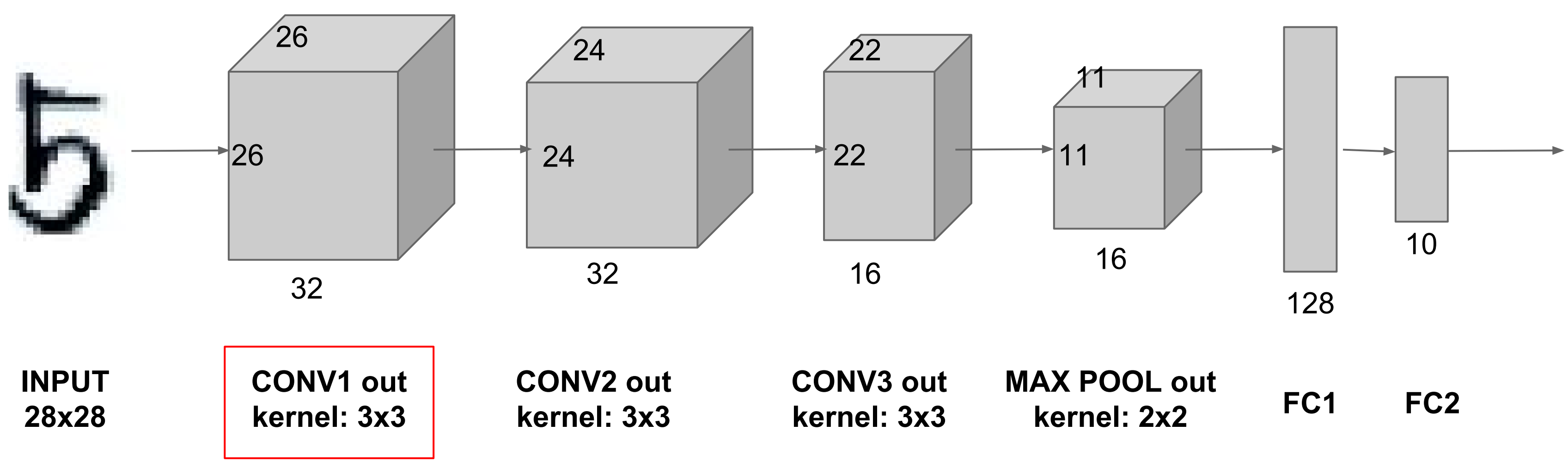
Visualization

Task 3

Visualization

Objective

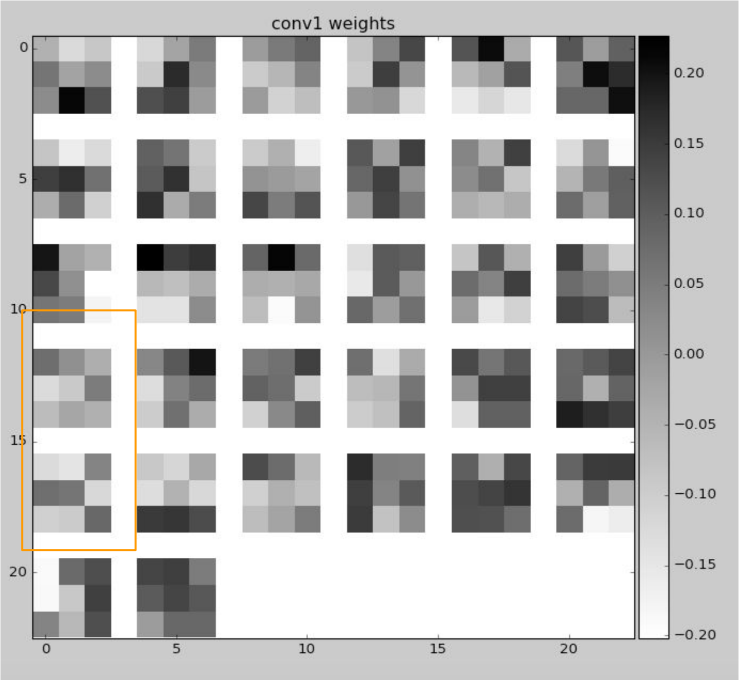
See the difference in the filter weights and activations through the process of training



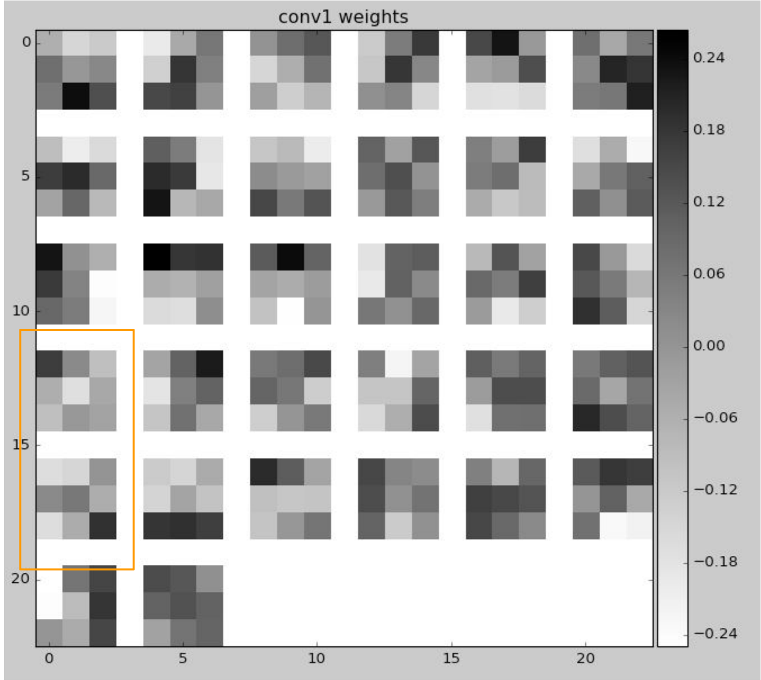
Task 3

Visualization

Convolutional Layer 1 - 32 filter weights



First epoch

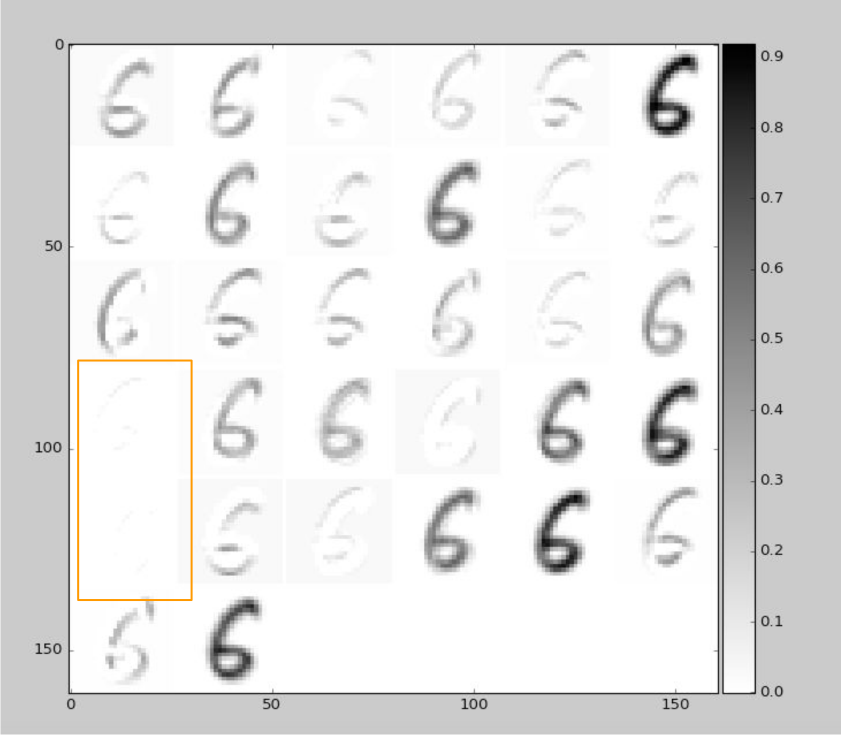


Last epoch

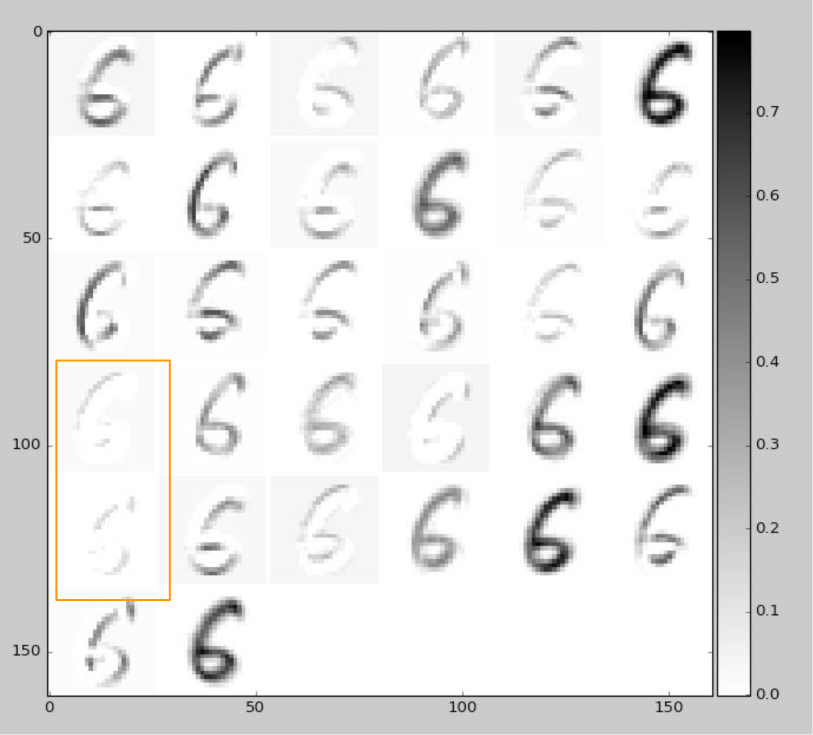
Task 3

Visualization

Convolutional Layer 1 - Activations



First epoch



Last epoch

Visualization

- No much change in the weights & activations due to the little loss of this specific dataset when training

- Allows us to see that each filter is focused in capturing different properties of the image (Edges, texture…)

Task 4

Transfer Learning

Task 4

Objective

- Fine-Tune on CIFAR-10

- Fine-Tune on VGG-16

We have the scripts ready to train!

However, loss and accuracy not decreasing

(We think that there is a data problem when loading the images or that they may not correspond with the labels)

We have not been able to finish that task.

(We did not have resources → Human, Computers)

Task 5

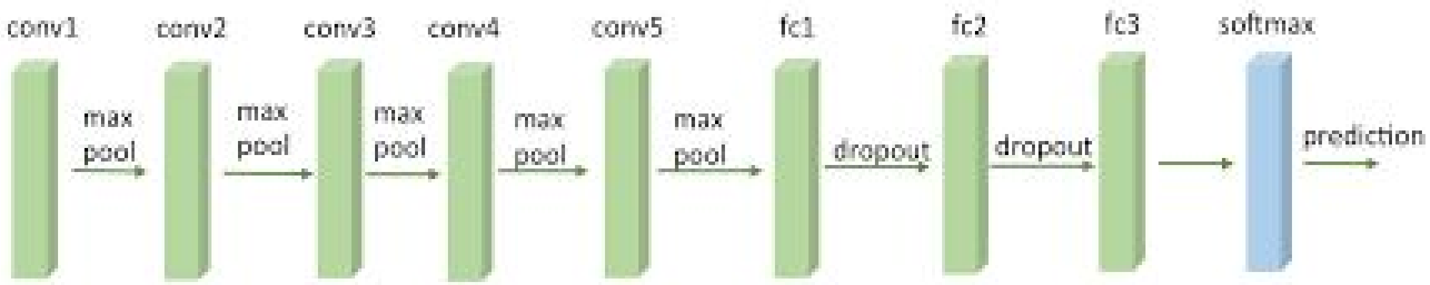
Deep Dream

Deep Dream

- VGG16

- 2 different transformations applied to different layers

- **Parameters**:
  - Continuity -> create artificial blur in the image
  - Dream -> L2 norm.(make image darker)
  - Jitter -> replacing each pixel with random pixel from neighborhood

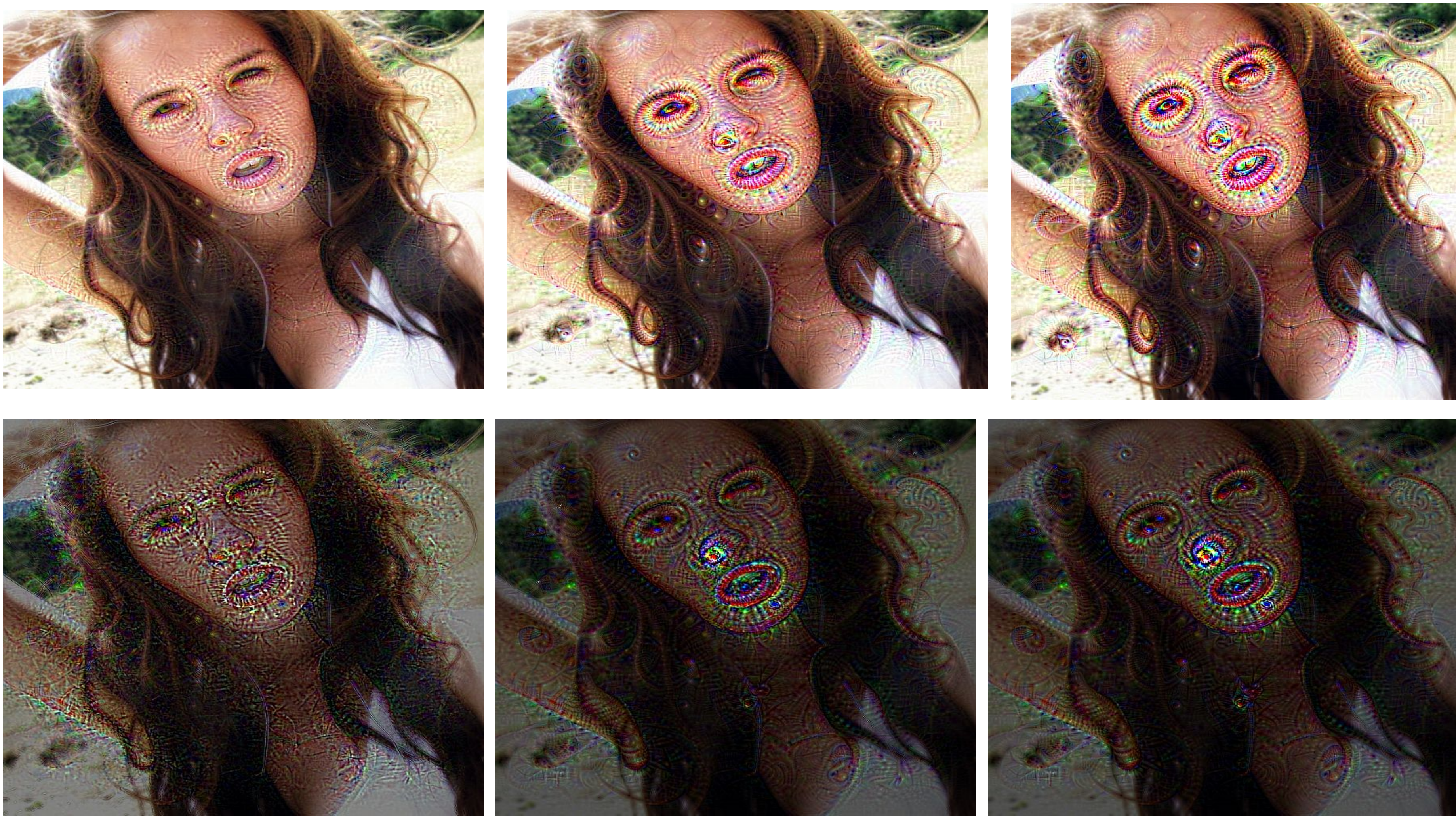


Deep Dream Results

- Different Conv Layers(2 & 5)

Deep Dream Results

- Same Layers(5), different settings(dream=0.8)



Conclusions

Conclusions

- We were very limited by not having computational resources.


THANK
YOU
GRACIAS
ARIGATO
SHUKURIA
DANKSCHEEN
BİYAN
SHUKRIA
TINGKI
YAQHANYELAY
TASHAKKUR ATU
SUKSAMA
EKHMET
MEHRBANI
GRAZIE
MAAKE
PALDIES
BOLZÏN
MERCI
GOZAIMASHITA
EFCHARISTO
JUSPAXAR
KOMAPSUMNIDA
SPASSIBO
SNACHALHUYA
NUHUN
CHALTU
WABEEJA
MAITEKA
YUSPAGARATAM
HUI
GUI
HATUR
UNALCHÉESH
SPASIBO
DENKAUJA
NENACHALHYA
MERSI
ATTO
DHANYAVAAD
SIKOMO
EKOJU
MAKETAI
MINMONCHAR
SANCO
MERASTAWHY
GAEJTHO
LAH
TAVTAPUCH
MEDAWAGSE
AGUYJE
FAKAAUE
BAIKA