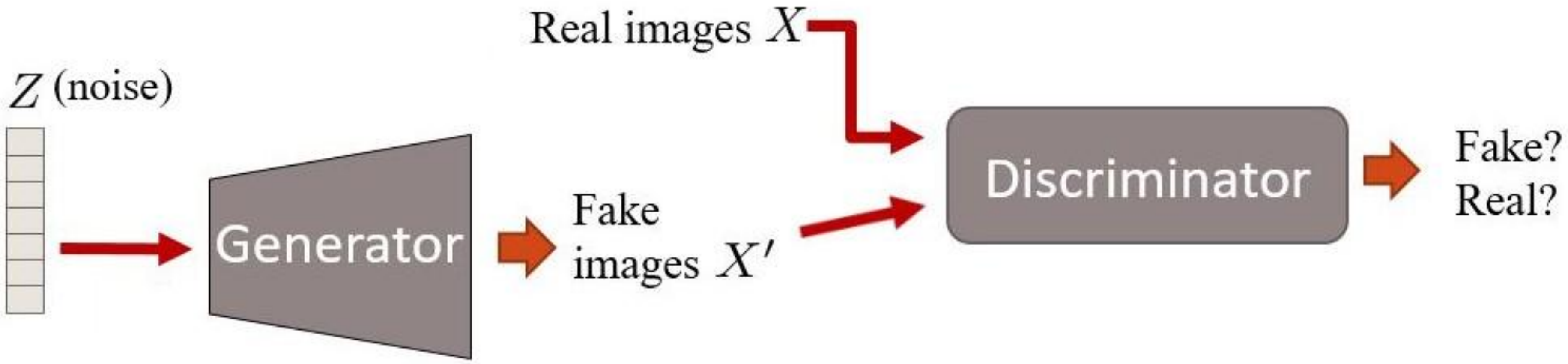
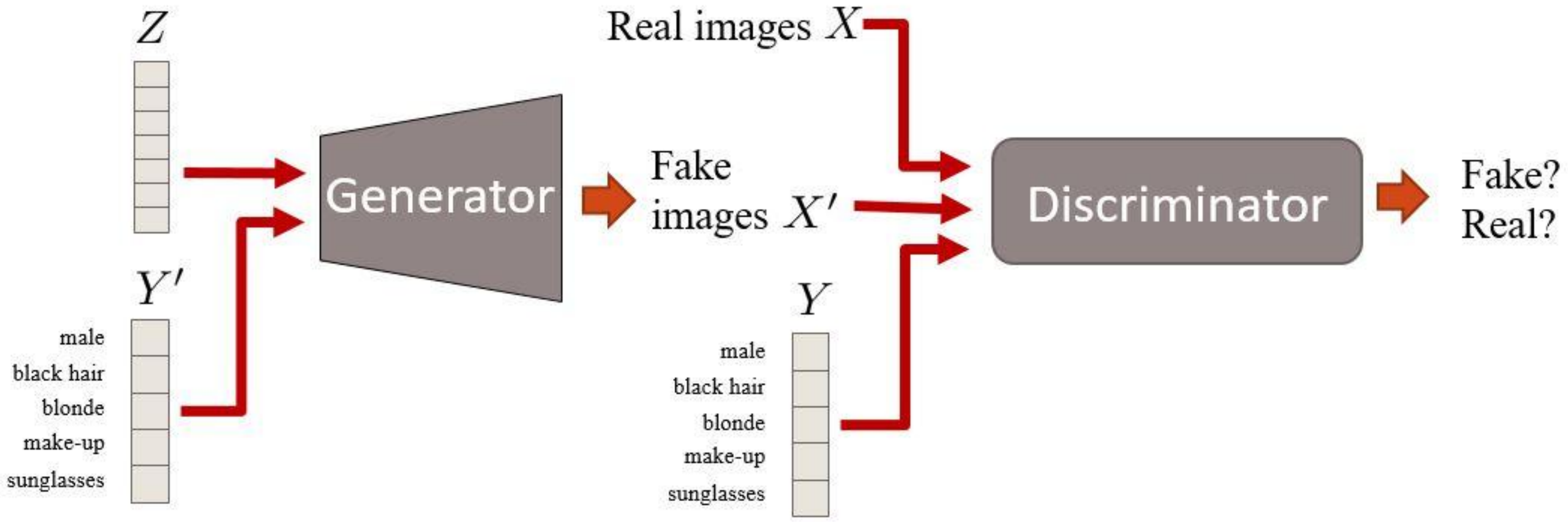
Conditional GAN Project

Filippo Bordon, Olivier Nicolini, Simone Zamboni

Normal GAN general architecture



Conditional GAN general architecture



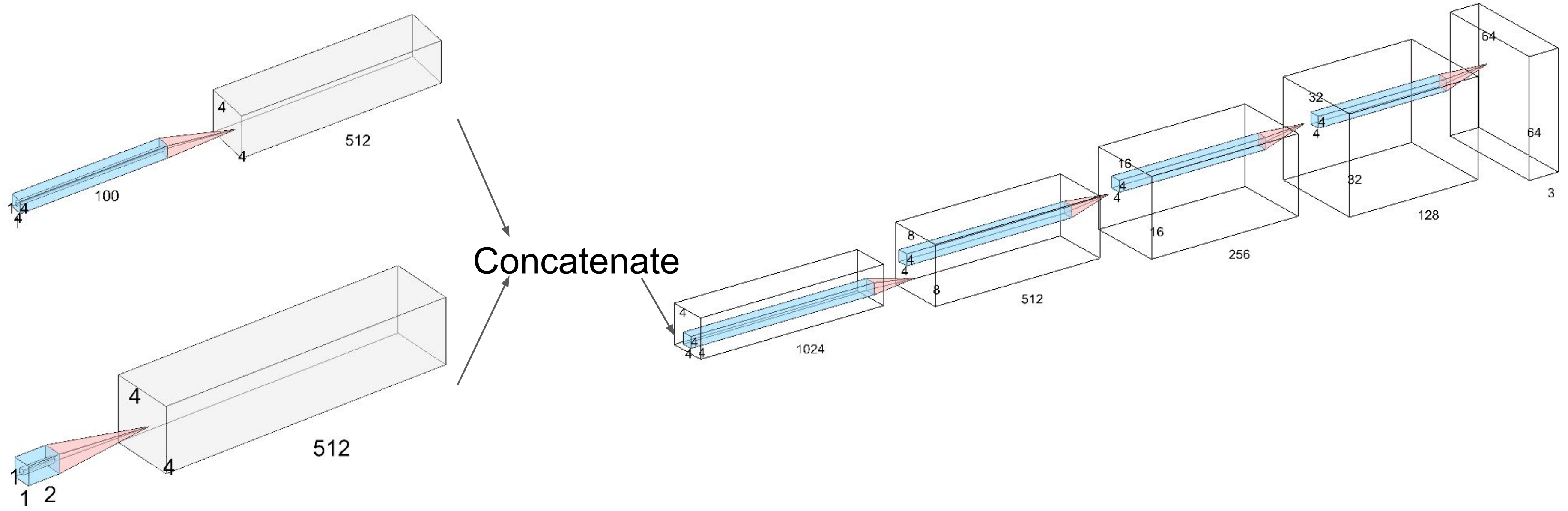
The Dataset: CelebA

For our test we used the CelebA dataset:

- More than 200k faces of famous people, mainly actors
- Cropped image of the faces, resized by 64x64
- 40 binary labels for each image, for example:
  - Male
  - Attractive
  - Blonde hair
  - Black hair
  - Young
  - Smiling
- Downloaded from Kaggle
- Challenging dataset: faces in very different poses

Baseline Generator architecture for cDCGAN

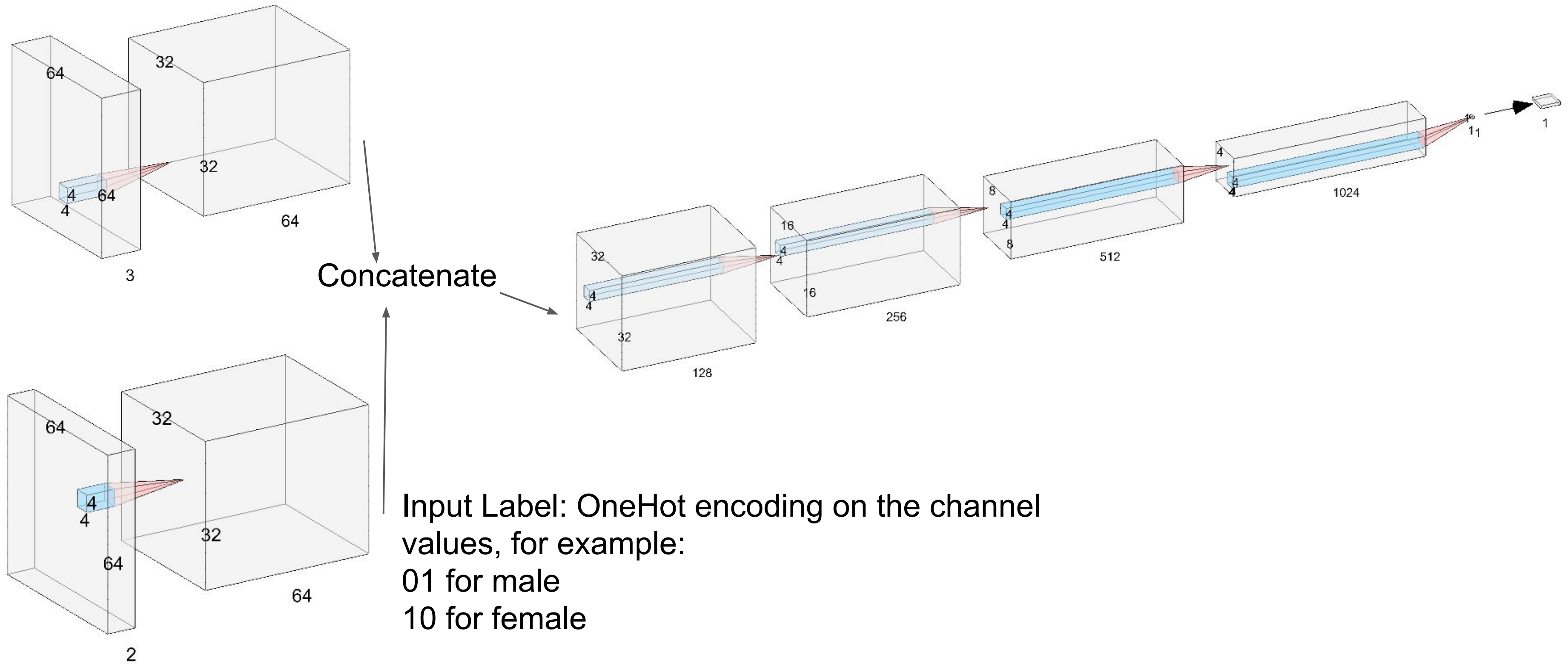
Noise input: a tensor 100x1x1



Label input, OneHot encoding, for example:
01 for male
10 for female

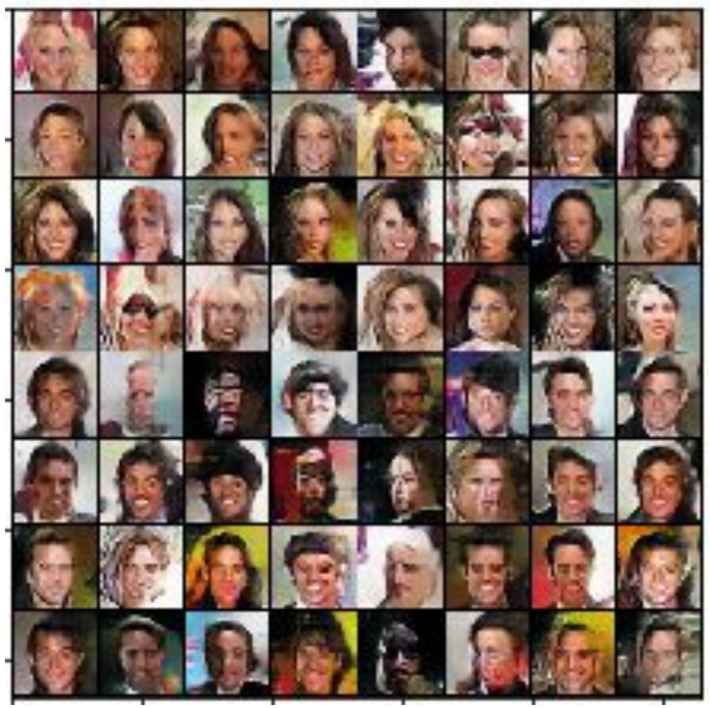
Baseline Discriminator Architecture for cDCGAN

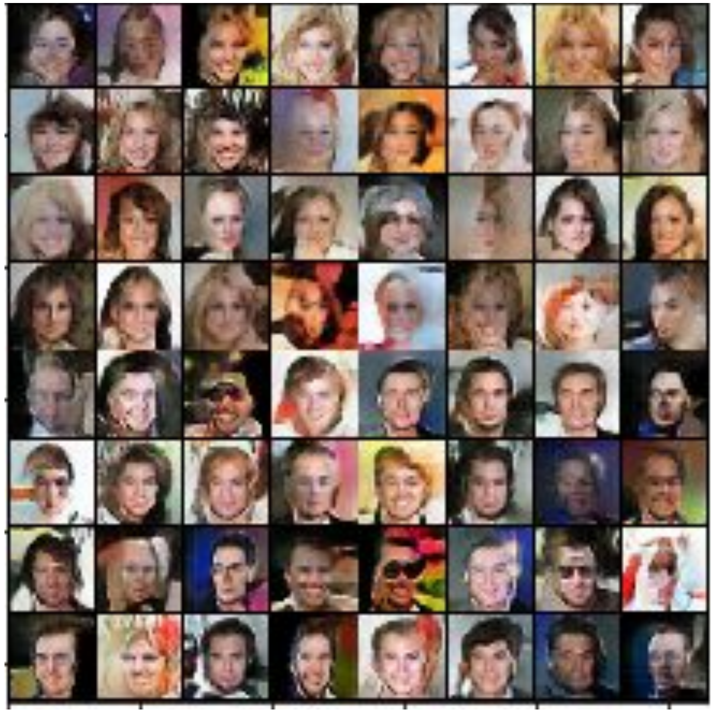
Input Image: 3 channel, 64x64 pixels



Input Label: OneHot encoding on the channel values, for example:
01 for male
10 for female

Base results (after 10 epochs) female/male

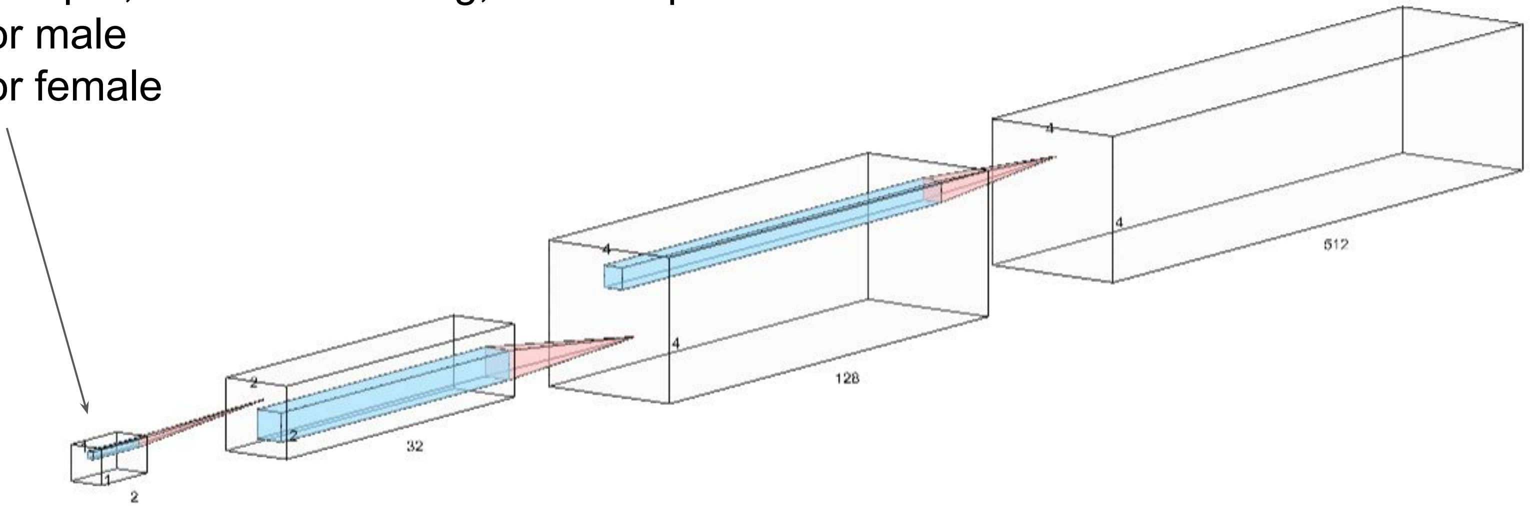




Change Architecture

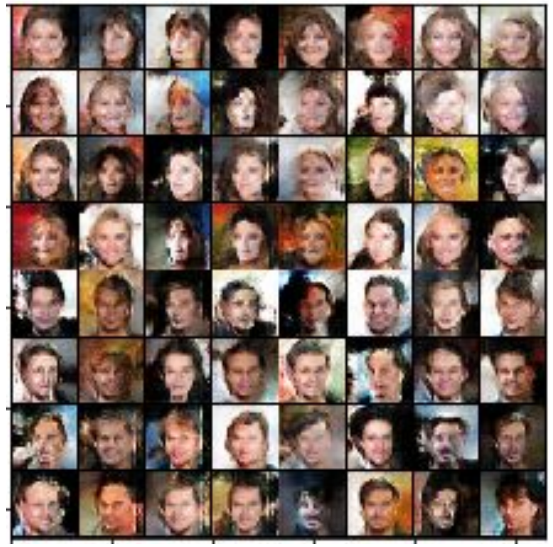
Add more convolutional layers for **label encoding** to the generator

Label input, OneHot encoding, for example:
01 for male
10 for female

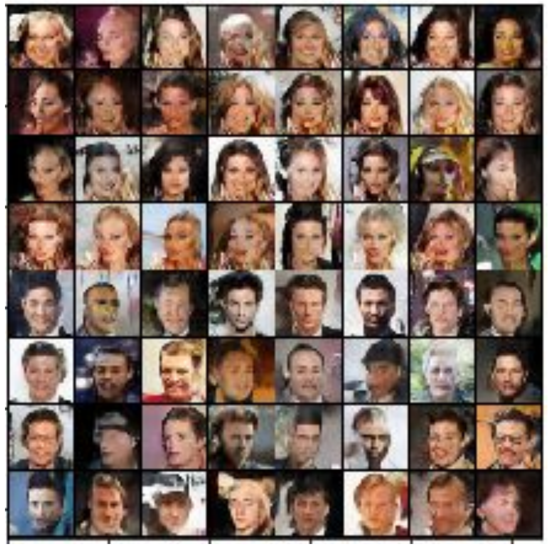


Changed architecture results

1st epoch



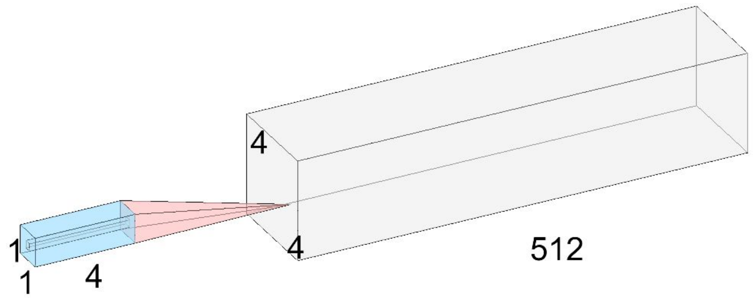
10th epoch



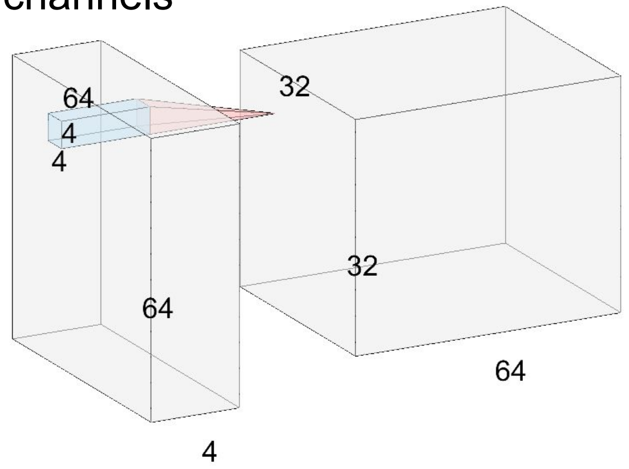
Better results, we can see faces from the first epoch and it already begins to show different faces for male and female

Can Multilabeling condition work?

Label in the generator for example:
0101: female and attractive
0110: female and non attractive
1001: male and attractive
1010: male and not attractive

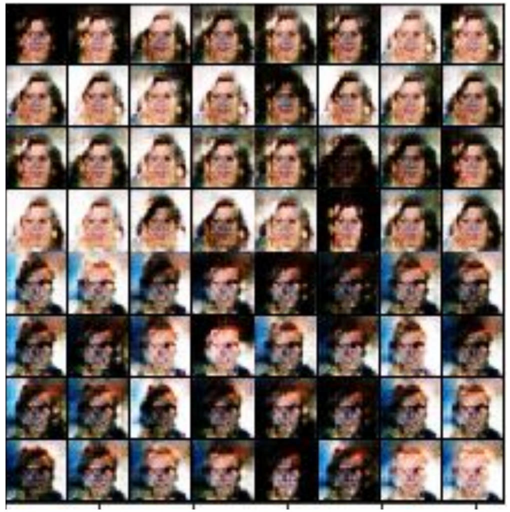


Label in the discriminator:
The same encoding of the generator but instead of single numbers we have channels

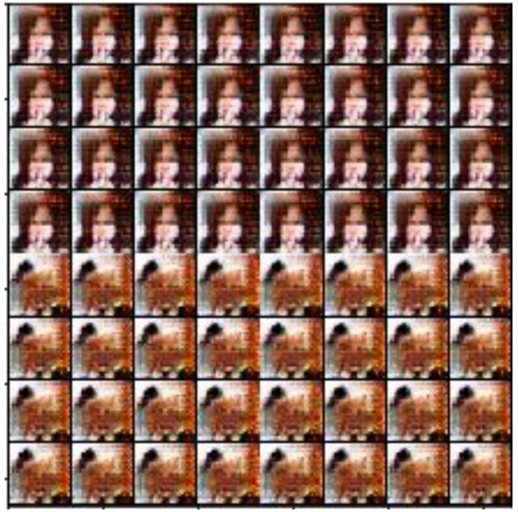


Results of multi-labeling for the base architecture, 10 epochs (pretty awful)

Labels: male/female and attractive/ non attractive (after 10 epochs)

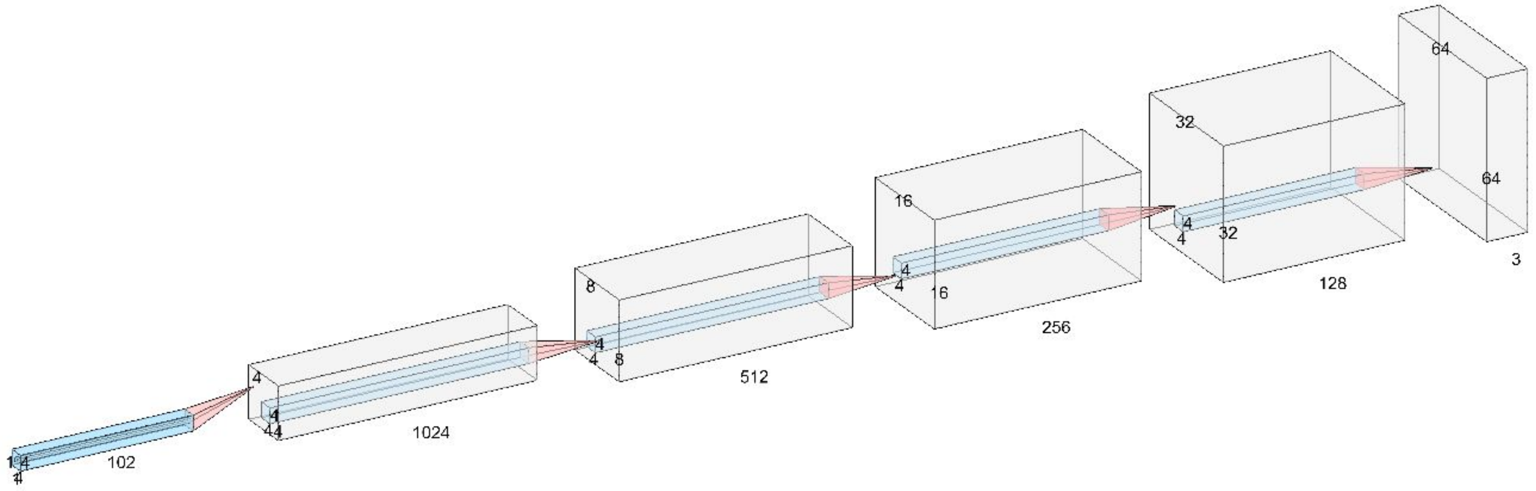


Labels: male/female, attractive/non attractive, black hair/ non black hair (after 10 epochs)



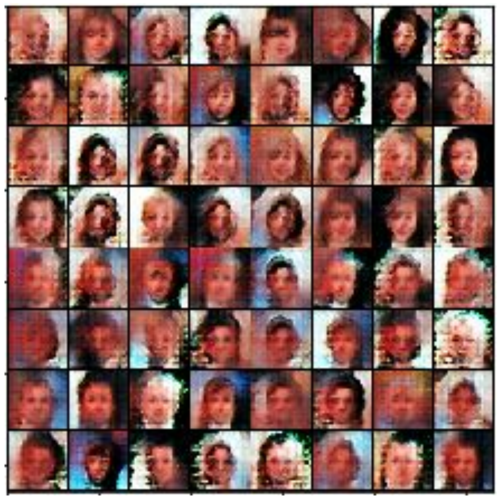
Can we embed the label information only in one input for the generator?

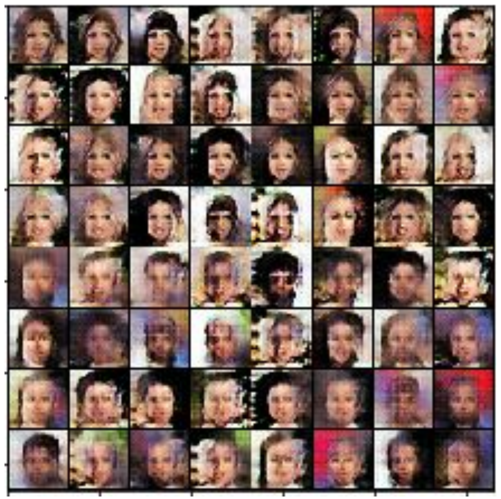
Modify the generator architecture: only one input label, 100 number for the noise input + 2 numbers which are a label using one-hot encoding



Results (10 epochs)

After 10 epochs the results are worse than the architecture with noise and label splitted.
Maybe there are too few layers for the label?



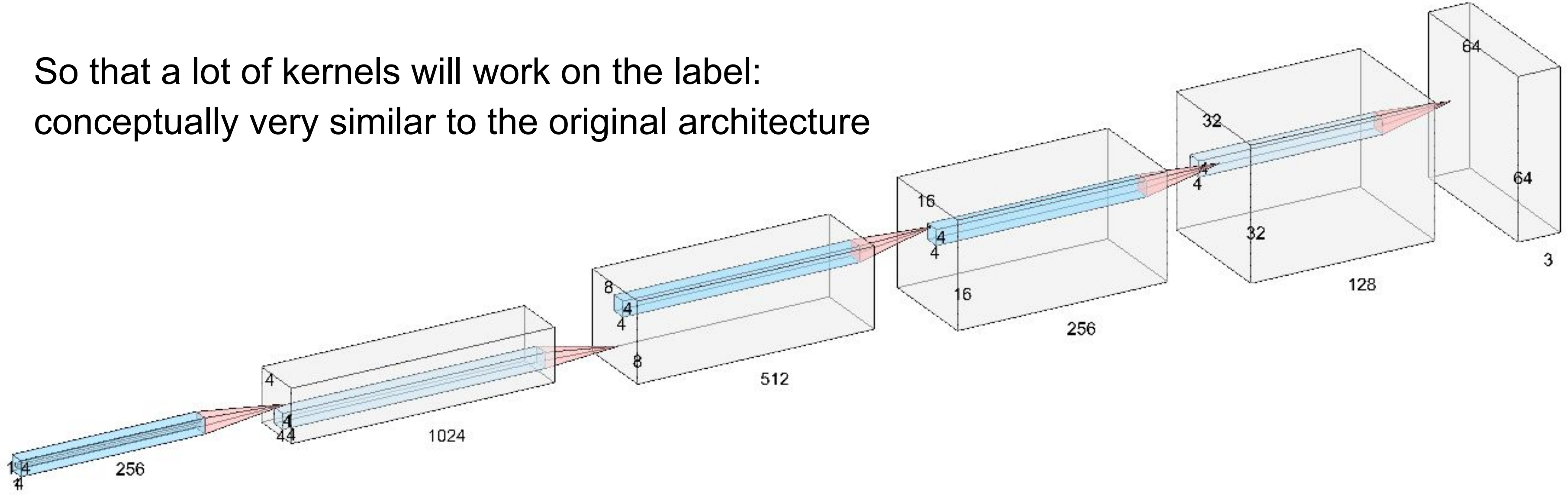


Label embedding in the noise 2

Let’s structure the input of the generator in this way:

- 128 number for the random noise +
- 128 numbers for the label: the first 64 one and the other 0 or vice versa

So that a lot of kernels will work on the label:
conceptually very similar to the original architecture



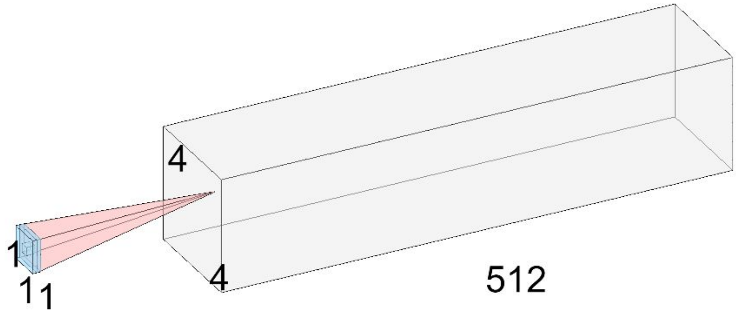
Results - 10 epochs (pretty awful)

Not that much better than the first embedding, maybe the label information is more present but is still way worse than the splitted one-hot encoding

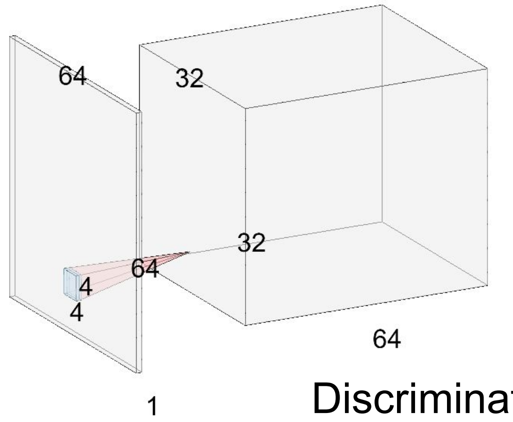
Different encoding than one-hot

Instead of a one-hot encoding we use a single number for a single label, for example:

- 0.5 female
- 1 male



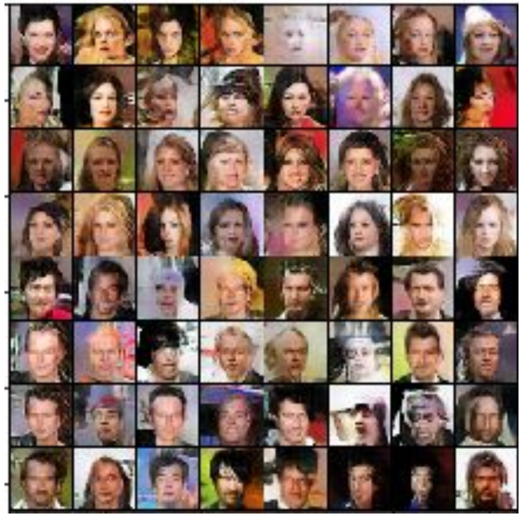
Generator

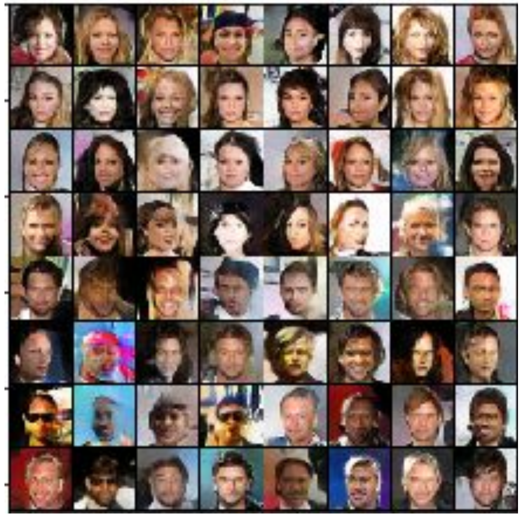


Discriminator

Results (10 epochs)

Results after 10 epochs are very similar to the one hot encoding





But the multilabel still doesn’t work...

Different encoding - Multilabel

Only one number as input for the label for the generator:

- 0.25 for female non black hair
- 0.5 for female black hair
- 0.75 for male non black hair
- 1 for male black hair

Different encoding - Multilabel

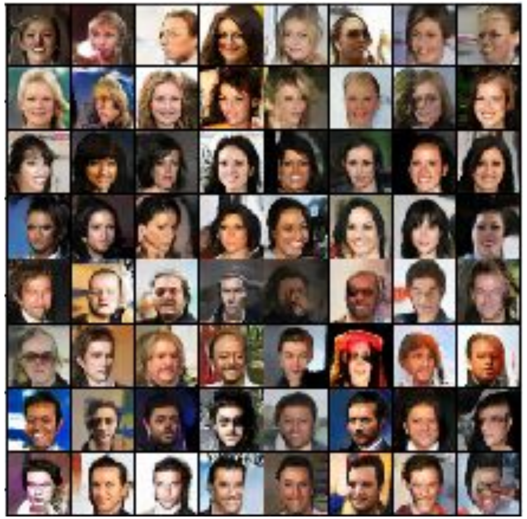
Results after 10 epochs for multilabels:

Female non Black hair

Female Black hair

Male non Black hair

Male Black hair



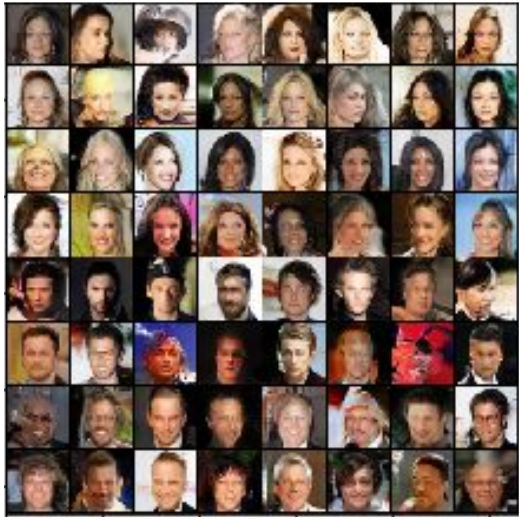
Labels: Black hair and Male

Female non smiling

Female smiling

Male non smiling

Male smiling



Labels: Smile and Male

Spectral normalization for multi-labeling

We tried to add spectral normalization to the **discriminator** in the architecture in order to **stabilize the training.**

Spectral normalization takes advantage of the Lipschitz continuity. It works by replacing every weight W with W/σ(W), so that it satisfies the Lipschitz constraint σ(W) = 1

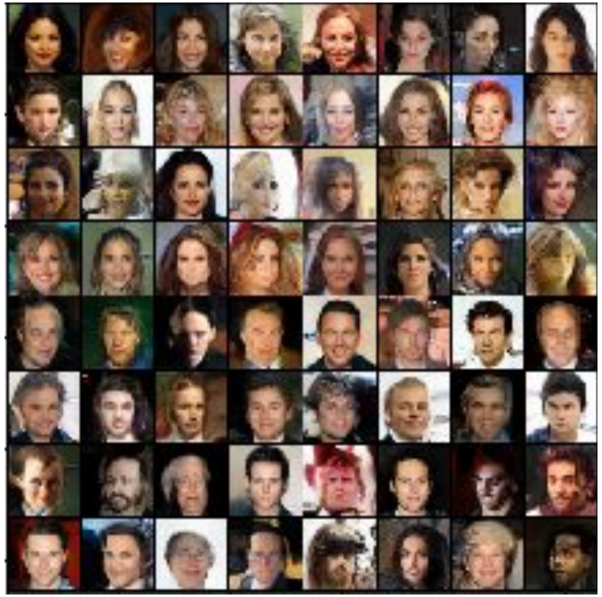
σ(A) is the spectral norm of the matrix A (L2 matrix norm of A), which is equivalent to the largest singular value of A

$$\sigma(A) := \max_{\boldsymbol{h}:\boldsymbol{h}\neq\boldsymbol{0}} \frac{\|A\boldsymbol{h}\|_2}{\|\boldsymbol{h}\|_2} = \max_{\|\boldsymbol{h}\|_2\leq 1} \|A\boldsymbol{h}\|_2,$$

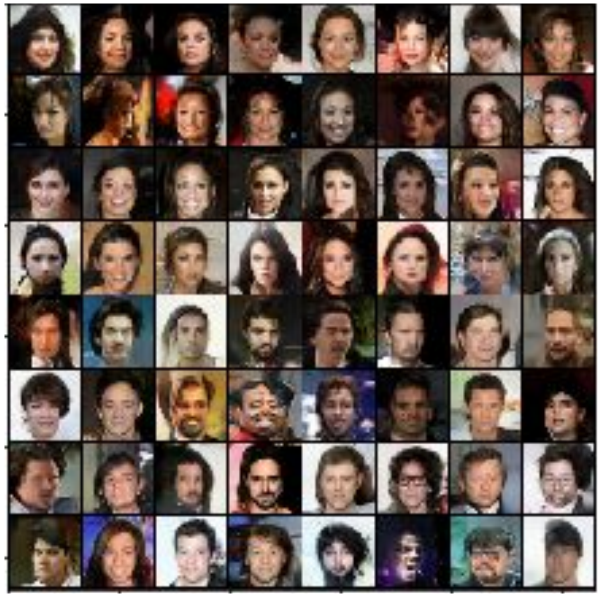
Spectral Normalization for Generative Adversarial Network - 2018 - T. Miyato , T. Kataoka , M. Koyama , Y. Yoshida

Spectral normalization results (10 epochs)

Male / Female



Male / Female + Black-hair / non Black hair



Female non Black hair

Female Black hair

Male non Black hair

Male Black hair

The results are a little bit better defined with respect to the previous case for just one attribute, but in the 2nd case the network is not able to create different images with the combination of the two labels, but instead creates images of men and women only with black hairs, but at least it converges.

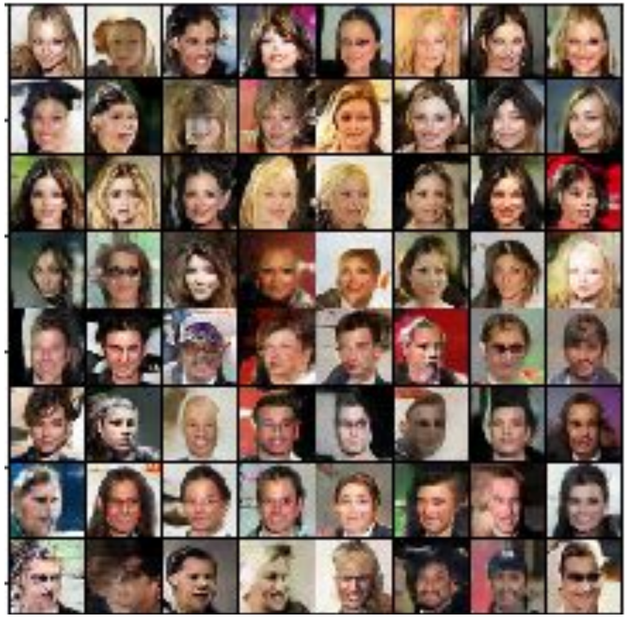
Categorical batch normalization

- We also try to add categorical batch normalization to the generator.
- Categorical batch normalization is a type of batch normalization that changes the normalization parameters based on the input category.
- More parameters → slower
- We have different batch normalization parameters for each label combination, for example:
    - female non black hair has its own normalization parameters
    - female black hair has its own normalization parameters
    - male non black hair has its own normalization parameters
    - male black hair has its own normalization parameters

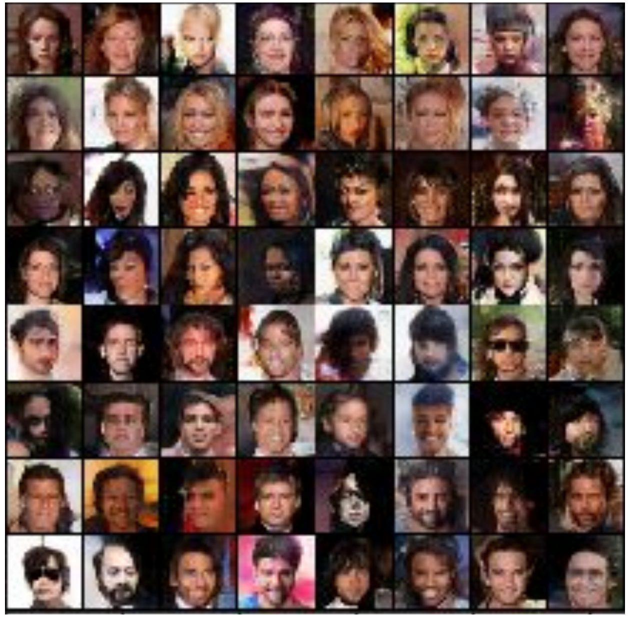
Modulating early visual processing by language - 2017 - H.de Vries, F.Strub, J.Mary, H.Larochelle, O.Pietquin, A.Courville

Categorical batch normalization results (10 epochs)

Male/Female



Male/Female and Black Hair / non Black Hair



Female non Black hair

Female Black hair

Male non Black hair

Male Black hair

Betters results, we are able to differentiate the results based on the combination of labels

Final architecture

- One hot encoding of the labels in vectors for the generator and in channels in the discriminator
- More convolutions of the label in the generator
- **Categorical batch normalization** in the generator
- **Spectral normalization** in the discriminator

Final results

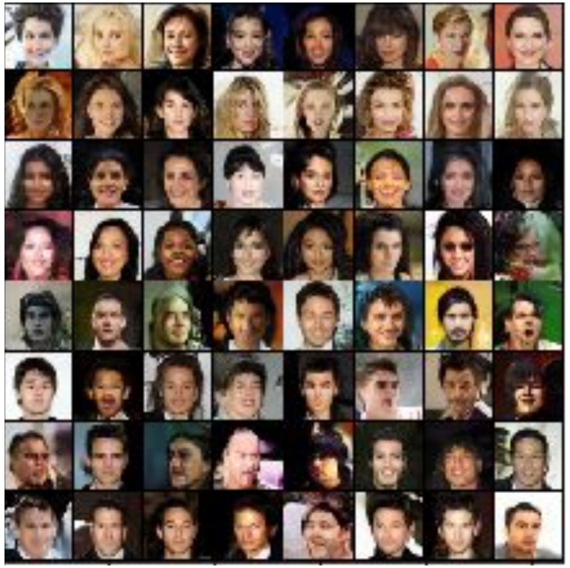
Male/Female and Black Hair / non Black Hair

Female non Black hair

Female Black hair

Male non Black hair

Male Black hair



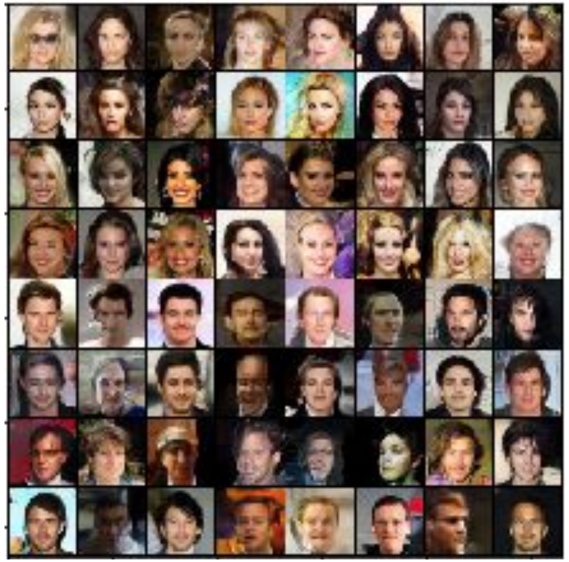
Male/Female and Smiling / non Smiling

Female non smiling

Female smiling

Male non smiling

Male smiling



Key Takeaway

- Training GANs is **very hard**
- When the algorithm doesn’t learn is because the loss of the discriminator evaluating the generated images is close to 0 and therefore the backpropagation changes little of the generator weights
- All labels are not created equally
- **Spectral normalization in the discriminator** helps
- **Categorical batch normalization** and more convolutions to the labels help in the generator

Conditional GAN Project

Filippo Bordon, Olivier Nicolini, Simone Zamboni