Generating Medical Image Data

Machine Learning for Health Informatics (LV 185.A83)

Marcus D. Bloice

Med. Univ. Graz

20 March 2018

About Me

A little bit about me (contact details are at the end of this presentation)

- I am a researcher at the Medical University Graz

About Me

A little bit about me (contact details are at the end of this presentation)

- I am a researcher at the Medical University Graz
- I work on applied machine learning in the healthcare domain, with a special interest in medical image data such as laser scanning microscopy data and histology image data

About Me

A little bit about me (contact details are at the end of this presentation)

- I am a researcher at the Medical University Graz
- I work on applied machine learning in the healthcare domain, with a special interest in medical image data such as laser scanning microscopy data and histology image data
- Have an interest in data augmentation and maintain a popular image augmentation software package called Augmentor, see `https://github.com/mdbloice/Augmentor`

About this Lecture

Today we are going to look at the **generation of image data**, and specifically medical image data. We will try to answer the following questions:

About this Lecture

Today we are going to look at the **generation of image data**, and specifically medical image data. We will try to answer the following questions:

- Why generate image data? Why generate *medical* image data?

About this Lecture

Today we are going to look at the **generation of image data**, and specifically medical image data. We will try to answer the following questions:

- Why generate image data? Why generate *medical* image data?
- How to generate image data?

About this Lecture

Today we are going to look at the **generation of image data**, and specifically medical image data. We will try to answer the following questions:

- Why generate image data? Why generate *medical* image data?
- How to generate image data?

Last we will discuss an **assigment** for the course: create a neural network that **generates** realistic, lifelike skin lesion images.

About this lecture

Note

These slides are quite descriptive and contain lots of text. This is by design! I have made them so that you can download them later, and read over even if you missed this lecture. So apologies if some of the slides seem overly verbose.

Procedure

We will split this talk into the following sections:

1. Motivation and Background

Procedure

We will split this talk into the following sections:

1. Motivation and Background
2. Generating Medical Data

Procedure

We will split this talk into the following sections:

1. Motivation and Background
2. Generating Medical Data
3. Generative Adversarial Networks

Procedure

We will split this talk into the following sections:

1. Motivation and Background
2. Generating Medical Data
3. Generative Adversarial Networks
4. Frameworks

Procedure

We will split this talk into the following sections:

1. Motivation and Background
2. Generating Medical Data
3. Generative Adversarial Networks
4. Frameworks
5. Project

Motivation and Background

Motivation

- Neural networks have been shown to be particularly good at image classification and object detection, voice recognition, translation, etc.

Motivation

- Neural networks have been shown to be particularly good at image classification and object detection, voice recognition, translation, etc.
- They are the best performing algorithms at many image detection tasks, and consistently win Kaggle competitions, outperforming many traditional computer vision methods

Motivation

- Neural networks have been shown to be particularly good at image classification and object detection, voice recognition, translation, etc.
- They are the best performing algorithms at many image detection tasks, and consistently win Kaggle competitions, outperforming many traditional computer vision methods
- Neural networks are basically essential for systems that rely on ablities that we as humans probably could not program, such as self-driving cars

Motivation

- Neural networks have been shown to be particularly good at image classification and object detection, voice recognition, translation, etc.
- They are the best performing algorithms at many image detection tasks, and consistently win Kaggle competitions, outperforming many traditional computer vision methods
- Neural networks are basically essential for systems that rely on ablities that we as humans probably could not program, such as self-driving cars

Motivation

- Neural networks have been shown to be particularly good at image classification and object detection, voice recognition, translation, etc.
- They are the best performing algorithms at many image detection tasks, and consistently win Kaggle competitions, outperforming many traditional computer vision methods
- Neural networks are basically essential for systems that rely on ablities that we as humans probably could not program, such as self-driving cars

In the past decade empirical evidence has shown that neural networks are a particularly useful tool. **Recent advancements have shown, however, that they can also be used to generate data.** That is the focus of this talk and the focus of an assignment for LV 185.A83!

Motivation

- Medical image analysis is a large field and neural network research in this field is growing fast

Motivation

- Medical image analysis is a large field and neural network research in this field is growing fast
- This is because in medicine there are several, real-world scenarios that could benefit greatly from neural networks

Motivation

- Medical image analysis is a large field and neural network research in this field is growing fast
- This is because in medicine there are several, real-world scenarios that could benefit greatly from neural networks
- New techniques in deep learning have used for breast cancer detection, skin lesion analysis, radiology diagnostic support, early diabetes detection, etc.

Motivation

- Medical image analysis is a large field and neural network research in this field is growing fast
- This is because in medicine there are several, real-world scenarios that could benefit greatly from neural networks
- New techniques in deep learning have used for breast cancer detection, skin lesion analysis, radiology diagnostic support, early diabetes detection, etc.
- However, because deep learning requires **relatively large amounts of data**, and because researchers outside of hospitals **often do not have access to enough medical data**, the field is probably being artificially hindered by a lack of access to sufficient amounts of image data

Motivation

Generating Medical Data

A somewhat new field has emerged which allows for data to be generated, using a special type of neural network called a *Generative Adversarial Network* (and variants of them).

These generative networks will be focus of this talk today. Your project work will consist of designing a generative network and creating new medical data.

Motivation

Why would you want to generate medical data? Why not just make more open data available to researchers?

- Medical data is sensitive, and there are many privacy issues

Motivation

Why would you want to generate medical data? Why not just make more open data available to researchers?

- Medical data is sensitive, and there are many privacy issues
- Therefore, it cannot normally be shared between researchers or institutions

Motivation

Why would you want to generate medical data? Why not just make more open data available to researchers?

- Medical data is sensitive, and there are many privacy issues
- Therefore, it cannot normally be shared between researchers or institutions
- Some open archives exist, but they are often small, toy examples

Motivation

Why would you want to generate medical data? Why not just make more open data available to researchers?

- Medical data is sensitive, and there are many privacy issues
- Therefore, it cannot normally be shared between researchers or institutions
- Some open archives exist, but they are often small, toy examples
- It can take years to get data access approved by ethics committees

Motivation

Why would you want to generate medical data? Why not just make more open data available to researchers?

- Medical data is sensitive, and there are many privacy issues
- Therefore, it cannot normally be shared between researchers or institutions
- Some open archives exist, but they are often small, toy examples
- It can take years to get data access approved by ethics committees
- **Deep neural networks** require relatively **large numbers of images to train**, especially if the task is complicated or nuanced (many classes, subtle differences: typical issues in the medical domain, such as in dermatology)

Motivation

However, there are **some** openly available medical image data sets.

- The idea for this assignment would be to collect such an open data set

Motivation

However, there are **some** openly available medical image data sets.

- The idea for this assignment would be to collect such an open data set
- Generate more images based on this existing data set

Motivation

However, there are **some** openly available medical image data sets.

- The idea for this assignment would be to collect such an open data set
- Generate more images based on this existing data set
- Evaluate whether generating such image data is ultimately useful by seeing if we can 'trick' dermatologists into thinking they are seeing real data

Motivation

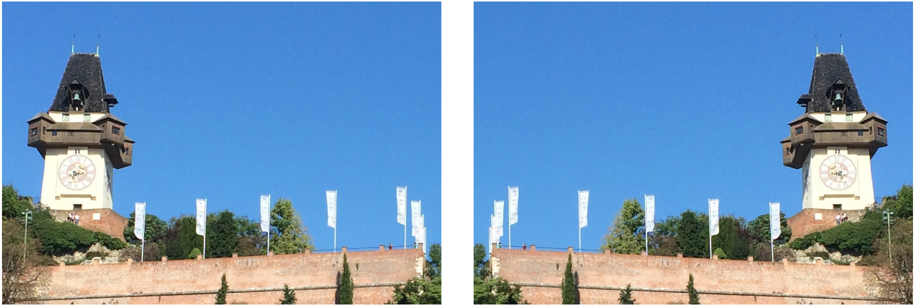
However, there are **some** openly available medical image data sets.

- The idea for this assignment would be to collect such an open data set
- Generate more images based on this existing data set
- Evaluate whether generating such image data is ultimately useful by seeing if we can ‘trick’ dermatologists into thinking they are seeing real data
- Crucially, any generated data are basically **novel, new data** and can be **shared with the community** for their research

Augmentation

Just as a side note, image generation is **a lot different** to image augmentation.

- Augmentation gets an image or images, and generates new data **based** on this data
- For example, you may have images of buildings, and flipping them horizontally, can double your data set size:



Motivation

So while augmentation may be useful, and certainly has benefits for training models and achieving better accuracy and generalisation, it is not useful for medical data that we may one day want to **share**.

Motivation

So while augmentation may be useful, and certainly has benefits for training models and achieving better accuracy and generalisation, it is not useful for medical data that we may one day want to **share**.

- The data created using augmentation **cannot** be shared or made available online without ethics approval

Motivation

So while augmentation may be useful, and certainly has benefits for training models and achieving better accuracy and generalisation, it is not useful for medical data that we may one day want to **share**.

- The data created using augmentation **cannot** be shared or made available online without ethics approval
- This is provided it could be anonymised in the first place, which is not something that can be guarenteed

Motivation

So while augmentation may be useful, and certainly has benefits for training models and achieving better accuracy and generalisation, it is not useful for medical data that we may one day want to **share**.

- The data created using augmentation **cannot** be shared or made available online without ethics approval
- This is provided it could be anonymised in the first place, which is not something that can be guarenteed
- If we **generate** data, this data can be shared with the research community

Generating Medical Data

Generating Medical Data: Generative Models

What is a generative model[1]?

- This is any model that uses a data set (such as a sample of images that represents a distribution) and attempts to learn an estimate of that distribution

[1] Overview of generative models: `https://blog.openai.com/generative-models/`

Generating Medical Data: Generative Models

What is a generative model[1]?

- This is any model that uses a data set (such as a sample of images that represents a distribution) and attempts to learn an estimate of that distribution
- There are several algorithms that build generative models, such as Autoencoders (specifically Variational Autoencoders), or Recurrent Neural Networks (specifically Pixel Recurrent Neural Networks[2])

[1] Overview of generative models: `https://blog.openai.com/generative-models/`

[2] See `https://arxiv.org/abs/1601.06759`

Generating Medical Data: Generative Models

What is a generative model[1]?

- This is any model that uses a data set (such as a sample of images that represents a distribution) and attempts to learn an estimate of that distribution
- There are several algorithms that build generative models, such as Autoencoders (specifically Variational Autoencoders), or Recurrent Neural Networks (specifically Pixel Recurrent Neural Networks[2])
- However, today we will take a look at **Generative Adversarial Networks** (GANs), and some variants such as **Deep Convolutional Generative Adversarial Networks** (DCGANs)

[1] Overview of generative models: `https://blog.openai.com/generative-models/`

[2] See `https://arxiv.org/abs/1601.06759`

Generating Medical Data: Generative Adverserial Networks

So how does a generative model work?

Generating Medical Data: Generative Adverserial Networks

So how does a generative model work?

- In 2014 a paper by Ian Goodfellow introduced *Generative Adversarial Networks* (GANs)[3]

[3] Goodfellow et al., **Generative Adversarial Networks**, *arXiv*, 2014
`https://arxiv.org/abs/1406.2661`

Generating Medical Data: Generative Adverserial Networks

So how does a generative model work?

- In 2014 a paper by Ian Goodfellow introduced *Generative Adversarial Networks* (GANs)[3]
- The basic idea is that you set **two** neural networks against each other

[3] Goodfellow et al., **Generative Adversarial Networks**, *arXiv*, 2014
`https://arxiv.org/abs/1406.2661`

Generating Medical Data: Generative Adverserial Networks

So how does a generative model work?

- In 2014 a paper by Ian Goodfellow introduced *Generative Adversarial Networks* (GANs)[3]
- The basic idea is that you set **two** neural networks against each other
- One network, the **generator**, generates new data

[3]Goodfellow et al., **Generative Adversarial Networks**, *arXiv*, 2014
`https://arxiv.org/abs/1406.2661`

Generating Medical Data: Generative Adverserial Networks

So how does a generative model work?

- In 2014 a paper by Ian Goodfellow introduced *Generative Adversarial Networks* (GANs)[3]
- The basic idea is that you set **two** neural networks against each other
- One network, the **generator**, generates new data
- The second network, the **discriminator**, evaluates the generated images

[3] Goodfellow et al., **Generative Adversarial Networks**, *arXiv*, 2014
`https://arxiv.org/abs/1406.2661`

Generating Medical Data: Generative Adverserial Networks

So how does a generative model work?

- In 2014 a paper by Ian Goodfellow introduced *Generative Adversarial Networks* (GANs)[3]
- The basic idea is that you set **two** neural networks against each other
- One network, the **generator**, generates new data
- The second network, the **discriminator**, evaluates the generated images
- You have an original data set which contains images that you want the network to generate

[3] Goodfellow et al., **Generative Adversarial Networks**, *arXiv*, 2014
`https://arxiv.org/abs/1406.2661`

Generating Medical Data: Generative Adverserial Networks

So how does a generative model work?

- In 2014 a paper by Ian Goodfellow introduced *Generative Adversarial Networks* (GANs)[3]
- The basic idea is that you set **two** neural networks against each other
- One network, the **generator**, generates new data
- The second network, the **discriminator**, evaluates the generated images
- You have an original data set which contains images that you want the network to generate
- The discriminator, when given an image, decides whether it is from this original data set or not!

[3] Goodfellow et al., **Generative Adversarial Networks**, *arXiv*, 2014 `https://arxiv.org/abs/1406.2661`

Generating Medical Data: Generative Adverserial Networks

So how does a generative model work?

- In 2014 a paper by Ian Goodfellow introduced *Generative Adversarial Networks* (GANs)[3]
- The basic idea is that you set **two** neural networks against each other
- One network, the **generator**, generates new data
- The second network, the **discriminator**, evaluates the generated images
- You have an original data set which contains images that you want the network to generate
- The discriminator, when given an image, decides whether it is from this original data set or not!
- The generator is trying to learn to makes images that the discriminator thinks is from the original data set!

[3]Goodfellow et al., **Generative Adversarial Networks**, *arXiv*, 2014 `https://arxiv.org/abs/1406.2661`

Generative Adversarial Networks

Often the following analogy is made using a cop and a counterfeiter:

- The counterfeiter is **learning to create fake Euro notes** to trick the cop

Generative Adversarial Networks

Often the following analogy is made using a cop and a counterfeiter:

- The counterfeiter is **learning to create fake Euro notes** to trick the cop
- The cop is **also learning to identify the fake Euro notes** to catch the counterfeiter

Generative Adversarial Networks

Often the following analogy is made using a cop and a counterfeiter:

- The counterfeiter is **learning to create fake Euro notes** to trick the cop
- The cop is **also learning to identify the fake Euro notes** to catch the counterfeiter
- This generates a type of feedback loop, the counterfeiter and the cop are trying to learn each other's methods

Generative Adversarial Networks

Often the following analogy is made using a cop and a counterfeiter:

- The counterfeiter is **learning to create fake Euro notes** to trick the cop
- The cop is **also learning to identify the fake Euro notes** to catch the counterfeiter
- This generates a type of feedback loop, the counterfeiter and the cop are trying to learn each other's methods

Generative Adversarial Networks

Often the following analogy is made using a cop and a counterfeiter:

- The counterfeiter is **learning to create fake Euro notes** to trick the cop
- The cop is **also learning to identify the fake Euro notes** to catch the counterfeiter
- This generates a type of feedback loop, the counterfeiter and the cop are trying to learn each other's methods

Remember

A GAN pits two neural networks against each other, with one (generator) trying to 'trick' the other (discriminator) into thinking it is seeing data from the original dataset. The goal of the generator is to create the fake images. The goal of the discriminator is to identify images as being fake or not.

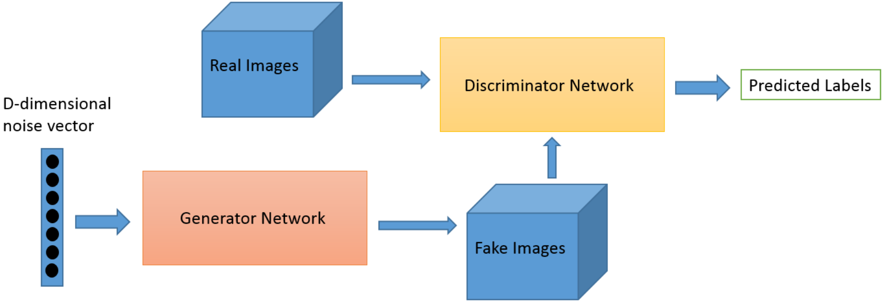
Generative Adversarial Networks

Greatly simplified, Generative Adversarial Networks are a type of **minimax** optimisation problem:

$$\min_G \max_D V(G, D)$$

You want to maximise the objective function of the discriminator, D, at correctly identifying the images as fake (or not). You also want to minimise the objective function of the generator, G, at generating images that are identified as real.

Generative Adversarial Networks



Credit: O'Reilly and
`https://deeplearning4j.org/generative-adversarial-network`

Generative Adversarial Networks

- The discriminator is a neural network that has the binary classification task of identifying images as real or fake

Generative Adversarial Networks

- The discriminator is a neural network that has the binary classification task of identifying images as real or fake
- We of course know whether the image we pass to the discriminator is real or fake!

Generative Adversarial Networks

- The discriminator is a neural network that has the binary classification task of identifying images as real or fake
- We of course know whether the image we pass to the discriminator is real or fake!
- The generator is also a neural network and it is trying to generate data that the discriminator classifies as real

Generative Adversarial Networks

- The discriminator is a neural network that has the binary classification task of identifying images as real or fake
- We of course know whether the image we pass to the discriminator is real or fake!
- The generator is also a neural network and it is trying to generate data that the discriminator classifies as real
- The generator starts with a low-dimensional noise vector (low compared to the original image data) so that the network is forced to learn a low-dimensional representation of the "essence" of the original data set. The internals of the network also have much fewer parameters than the original image data.

Generative Adversarial Networks

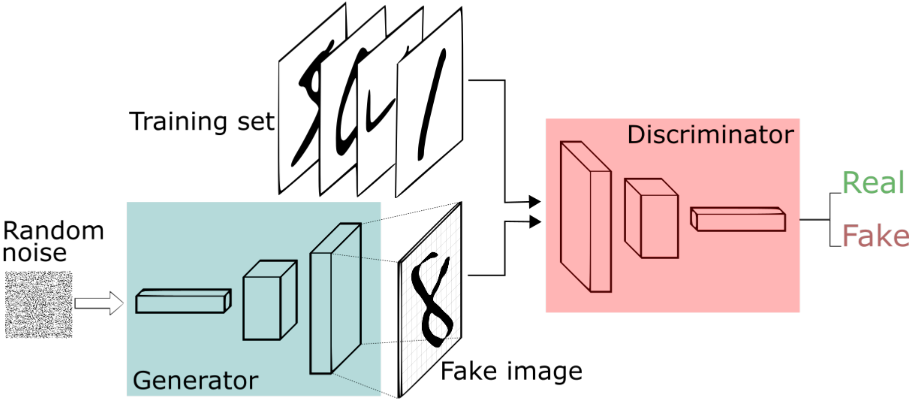
- The discriminator is a neural network that has the binary classification task of identifying images as real or fake
- We of course know whether the image we pass to the discriminator is real or fake!
- The generator is also a neural network and it is trying to generate data that the discriminator classifies as real
- The generator starts with a low-dimensional noise vector (low compared to the original image data) so that the network is forced to learn a low-dimensional representation of the "essence" of the original data set. The internals of the network also have much fewer parameters than the original image data.

Generative Adversarial Networks

- The discriminator is a neural network that has the binary classification task of identifying images as real or fake
- We of course know whether the image we pass to the discriminator is real or fake!
- The generator is also a neural network and it is trying to generate data that the discriminator classifies as real
- The generator starts with a low-dimensional noise vector (low compared to the original image data) so that the network is forced to learn a low-dimensional representation of the "essence" of the original data set. The internals of the network also have much fewer parameters than the original image data.

The previous slide depicts a general description of a GAN. Many types exist, including Deep Convolutional GANs, or DCGANs.

Generative Adversarial Networks



Credit:
https://deeplearning4j.org/generative-adversarial-network

Generative Adversarial Networks

- As mentioned previously **data generation** is different to **data augmentation** (where we just alter the original samples to create "new" samples)

Generative Adversarial Networks

- As mentioned previously **data generation** is different to **data augmentation** (where we just alter the original samples to create "new" samples)
- We are in fact creating data from the same distribution as the original data set

Generative Adversarial Networks

- As mentioned previously **data generation** is different to **data augmentation** (where we just alter the original samples to create "new" samples)
- We are in fact creating data from the same distribution as the original data set
- So although we are creating data from the same distribution of the original data this is **new** data

Generative Adversarial Networks

- As mentioned previously **data generation** is different to **data augmentation** (where we just alter the original samples to create "new" samples)
- We are in fact creating data from the same distribution as the original data set
- So although we are creating data from the same distribution of the original data this is **new** data
- Unlike augmentation we are **able to share this data with the research community**

Generative Adversarial Networks

- As mentioned previously **data generation** is different to **data augmentation** (where we just alter the original samples to create "new" samples)
- We are in fact creating data from the same distribution as the original data set
- So although we are creating data from the same distribution of the original data this is **new** data
- Unlike augmentation we are **able to share this data with the research community**
- Our aim therefore in this assignment is to create medical image data that should be realistic enough to be useful (from a research standpoint) and can be shared online without privacy concerns

Applications of Generative Adversarial Networks

Let's have a look at some samples of generative adversarial networks in use.

- Image generation

Applications of Generative Adversarial Networks

Let's have a look at some samples of generative adversarial networks in use.

- Image generation
- Increasing the resolution of images

Applications of Generative Adversarial Networks

Let's have a look at some samples of generative adversarial networks in use.

- Image generation
- Increasing the resolution of images
- Interactive image generation

Applications of Generative Adversarial Networks

Let's have a look at some samples of generative adversarial networks in use.

- Image generation
- Increasing the resolution of images
- Interactive image generation
- Image to image translation

Applications of Generative Adversarial Networks

Let's have a look at some samples of generative adversarial networks in use.

- Image generation
- Increasing the resolution of images
- Interactive image generation
- Image to image translation
- Text to image translation

Applications of Generative Adversarial Networks

Radford, Alec, et al. **Unsupervised representation learning with deep convolutional generative adversarial networks**. *arXiv preprint arXiv:1511.06434* (2015). See `https://github.com/Newmu/dcgan_code`

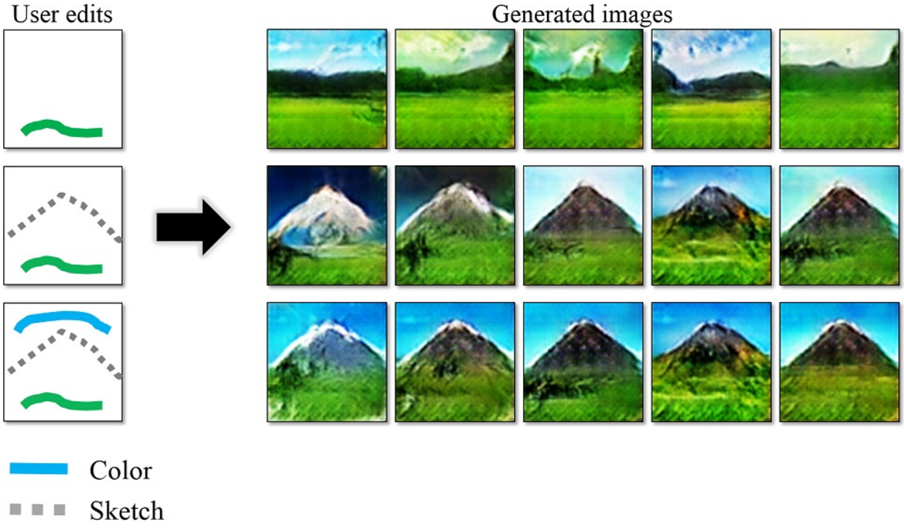
Applications of Generative Adversarial Networks

Radford, Alec, et al. **Unsupervised representation learning with deep convolutional generative adversarial networks**. *arXiv preprint arXiv:1511.06434* (2015). See `https://github.com/Newmu/dcgan_code`

Applications of Generative Adversarial Networks

Ledig, Christian, et al. **Photo-realistic single image super-resolution using a generative adversarial network**. *arXiv preprint arXiv:1609.04802* (2016).

Applications of Generative Adversarial Networks



Applications of Generative Adversarial Networks

Isola et al., **Image-to-Image Translation with Conditional Adversarial Networks**, *arXiv:1611.07004* (2016)

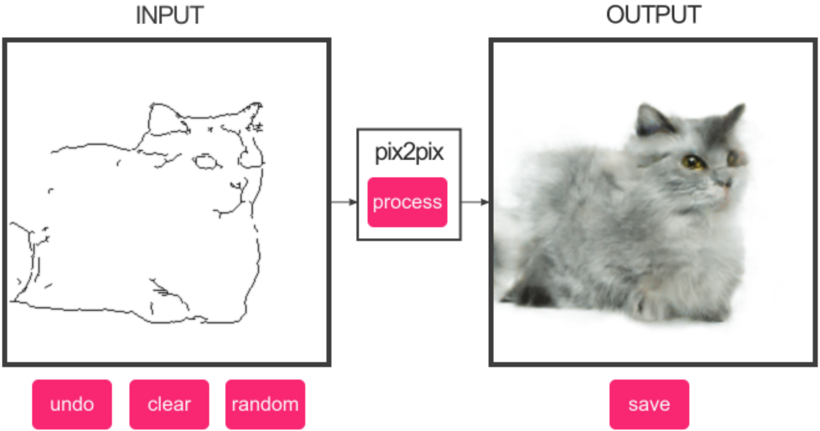
Applications of Generative Adversarial Networks

Isola et al., **Image-to-Image Translation with Conditional Adversarial Networks**, *arXiv:1611.07004* (2016)

Applications of Generative Adversarial Networks

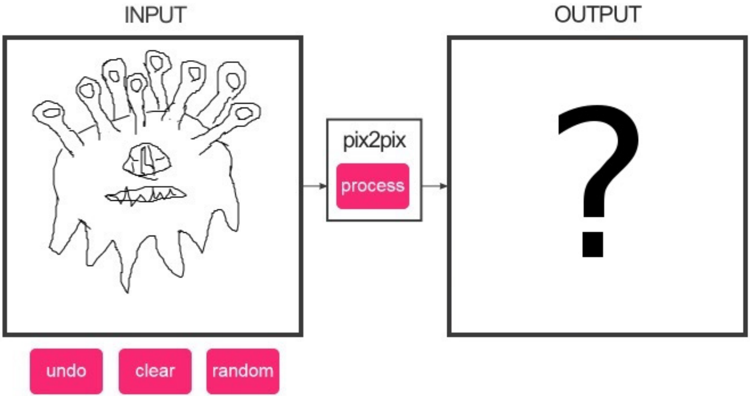
Isola et al., **Image-to-Image Translation with Conditional Adversarial Networks**, *arXiv:1611.07004* (2016)

Applications of Generative Adversarial Networks



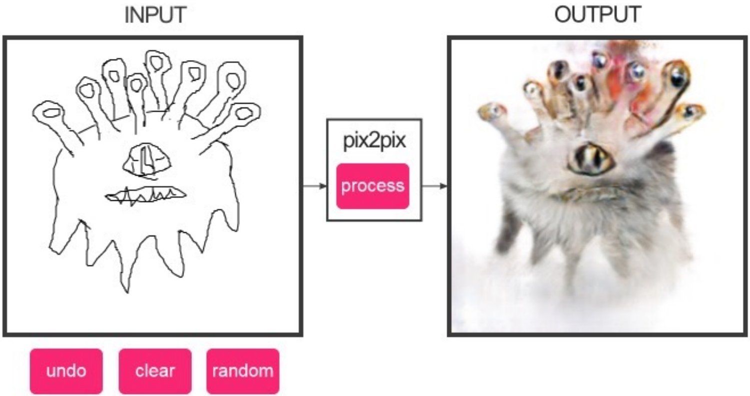
See `https://affinelayer.com/pixsrv/` and
`https://phillipi.github.io/pix2pix/`

Applications of Generative Adversarial Networks



See `https://affinelayer.com/pixsrv/` and `https://phillipi.github.io/pix2pix/`

Applications of Generative Adversarial Networks



See `https://affinelayer.com/pixsrv/` and
`https://phillipi.github.io/pix2pix/`

Applications of Generative Adversarial Networks

Flower has white petals and a yellow stamen

Reed et al., **Generative Adversarial Text to Image Synthesis**, *arXiv:1605.05396* (2016) `https://arxiv.org/abs/1605.05396` and `https://github.com/reedscot/icml2016`

Frameworks

Implementing a Generative Adversarial Network

Due to the popularity of deep learning and neural network based models, there are unsurprisingly a large amount of frameworks available.

- This is good because you have a lot of options for building your neural networks

Implementing a Generative Adversarial Network

Due to the popularity of deep learning and neural network based models, there are unsurprisingly a large amount of frameworks available.

- This is good because you have a lot of options for building your neural networks
- Some frameworks are **low-level**, while others are **high-level** and based on some low-level framework

Implementing a Generative Adversarial Network

Due to the popularity of deep learning and neural network based models, there are unsurprisingly a large amount of frameworks available.

- This is good because you have a lot of options for building your neural networks
- Some frameworks are **low-level**, while others are **high-level** and based on some low-level framework
- Low-level frameworks are general symbolic math libraries or numerical computation and matrix libraries

Implementing a Generative Adversarial Network

Due to the popularity of deep learning and neural network based models, there are unsurprisingly a large amount of frameworks available.

- This is good because you have a lot of options for building your neural networks
- Some frameworks are **low-level**, while others are **high-level** and based on some low-level framework
- Low-level frameworks are general symbolic math libraries or numerical computation and matrix libraries
- High-level frameworks offer many convenience functions and let you build algorithms using an API

Implementing a Generative Adversarial Network

Due to the popularity of deep learning and neural network based models, there are unsurprisingly a large amount of frameworks available.

- This is good because you have a lot of options for building your neural networks
- Some frameworks are **low-level**, while others are **high-level** and based on some low-level framework
- Low-level frameworks are general symbolic math libraries or numerical computation and matrix libraries
- High-level frameworks offer many convenience functions and let you build algorithms using an API
- Today we are only going to talk about **Python-based frameworks**

Keras

- Keras is a **high-level neural networks API**

Keras

- Keras is a **high-level neural networks API**
- Written in Python, install it via pip (`pip install keras`)

Keras

- Keras is a **high-level neural networks API**
- Written in Python, install it via pip (`pip install keras`)
- It supports several back ends such as **TensorFlow**, Theano, and CNKT

Keras

- Keras is a **high-level neural networks API**
- Written in Python, install it via pip (`pip install keras`)
- It supports several back ends such as **TensorFlow**, Theano, and CNKT
- Supports the GPU (or CPU, of course)

Keras

- Keras is a **high-level neural networks API**
- Written in Python, install it via pip (`pip install keras`)
- It supports several back ends such as **TensorFlow**, Theano, and CNKT
- Supports the GPU (or CPU, of course)

Keras

- Keras is a **high-level neural networks API**
- Written in Python, install it via pip (`pip install keras`)
- It supports several back ends such as **TensorFlow**, Theano, and CNKT
- Supports the GPU (or CPU, of course)

Sample source code:

```
model = Sequential()
model.add(Dense(n_classes, input_dim=n_dims))
model.add(Activation('softmax'))
model.compile(optimizer='sgd',
              loss='categorical_crossentropy')
model.fit(X_train, y_train)
```

This creates and trains a simple, single layer neural network.

Theano

Theano, on the other hand, is an example of a low-level library:

```
W = theano.shared(
    value=numpy.zeros(
        (n_in, n_out),
        dtype=theano.config.floatX
    ),
    name='W',
    borrow=True
)
# Define biases vector b
...

p_y_given_x = T.nnet.softmax(T.dot(input, W) + b)
y_pred = T.argmax(p_y_given_x, axis=1)
```

See `http://deeplearning.net/tutorial/logreg.html`

PyTorch

- Torch is a framework which is used and developed by Facebook

PyTorch

- Torch is a framework which is used and developed by Facebook
- Originally written in Lua

PyTorch

- Torch is a framework which is used and developed by Facebook
- Originally written in Lua
- However, PyTorch allows you to use Python for development:

PyTorch

- Torch is a framework which is used and developed by Facebook
- Originally written in Lua
- However, PyTorch allows you to use Python for development:

PyTorch

- Torch is a framework which is used and developed by Facebook
- Originally written in Lua
- However, PyTorch allows you to use Python for development:

```
import torch.nn as nn
# Make your own class inheriting from Module
class LeNet5(nn.Module):
    def __init__(self):
        super(LeNet5, self).__init__()
        self.conv1 = nn.Conv2d(3, 6, 5)
        self.conv2 = nn.Conv2d(6, 16, 5)
        self.fc1   = nn.Linear(16*5*5, 120)
        ...
```

- **Linux/UNIX only!**

TensorFlow

- TensorFlow is Developed by Google, and is quite low-level (but is installed as a Python package)

TensorFlow

- TensorFlow is Developed by Google, and is quite low-level (but is installed as a Python package)
- We already mentioned that TensorFlow is the default back-end for Keras

TensorFlow

- TensorFlow is Developed by Google, and is quite low-level (but is installed as a Python package)
- We already mentioned that TensorFlow is the default back-end for Keras
- For this assignment I would recommend Keras + TensorFlow

TensorFlow

- TensorFlow is Developed by Google, and is quite low-level (but is installed as a Python package)
- We already mentioned that TensorFlow is the default back-end for Keras
- For this assignment I would recommend Keras + TensorFlow
- Easy to install, even for the GPU version:

TensorFlow

- TensorFlow is Developed by Google, and is quite low-level (but is installed as a Python package)
- We already mentioned that TensorFlow is the default back-end for Keras
- For this assignment I would recommend Keras + TensorFlow
- Easy to install, even for the GPU version:

TensorFlow

- TensorFlow is Developed by Google, and is quite low-level (but is installed as a Python package)
- We already mentioned that TensorFlow is the default back-end for Keras
- For this assignment I would recommend Keras + TensorFlow
- Easy to install, even for the GPU version:

```
$ pip install tensorflow-gpu
```

TensorFlow

- TensorFlow is Developed by Google, and is quite low-level (but is installed as a Python package)
- We already mentioned that TensorFlow is the default back-end for Keras
- For this assignment I would recommend Keras + TensorFlow
- Easy to install, even for the GPU version:

```
$ pip install tensorflow-gpu
```

- TensorFlow is a general, numerical computation library

TensorFlow

- TensorFlow is Developed by Google, and is quite low-level (but is installed as a Python package)
- We already mentioned that TensorFlow is the default back-end for Keras
- For this assignment I would recommend Keras + TensorFlow
- Easy to install, even for the GPU version:

```
$ pip install tensorflow-gpu
```

- TensorFlow is a general, numerical computation library
- You can use TensorFlow to implement random forests, K-means, etc. as well as deep neural networks

TensorFlow

- TensorFlow is Developed by Google, and is quite low-level (but is installed as a Python package)
- We already mentioned that TensorFlow is the default back-end for Keras
- For this assignment I would recommend Keras + TensorFlow
- Easy to install, even for the GPU version:

```
$ pip install tensorflow-gpu
```

- TensorFlow is a general, numerical computation library
- You can use TensorFlow to implement random forests, K-means, etc. as well as deep neural networks

TensorFlow

- TensorFlow is Developed by Google, and is quite low-level (but is installed as a Python package)
- We already mentioned that TensorFlow is the default back-end for Keras
- For this assignment I would recommend Keras + TensorFlow
- Easy to install, even for the GPU version:

```
$ pip install tensorflow-gpu
```

- TensorFlow is a general, numerical computation library
- You can use TensorFlow to implement random forests, K-means, etc. as well as deep neural networks

A very good collection of examples can be found here:
https://github.com/aymericdamien/TensorFlow-Examples
(includes an example GAN, and many more)

Notes on Frameworks

- As stated previosuly, for this assignment I would recommend Keras using TensorFlow as a back-end

Notes on Frameworks

- As stated previosuly, for this assignment I would recommend Keras using TensorFlow as a back-end
- You can work exclusively in the high-level Keras environment

Notes on Frameworks

- As stated previosuly, for this assignment I would recommend Keras using TensorFlow as a back-end
- You can work exclusively in the high-level Keras environment
- Both are in active development with large user bases so there is much help available online

Notes on Frameworks

- As stated previosuly, for this assignment I would recommend Keras using TensorFlow as a back-end
- You can work exclusively in the high-level Keras environment
- Both are in active development with large user bases so there is much help available online
- Both are easy to install using `pip`:

Notes on Frameworks

- As stated previosuly, for this assignment I would recommend Keras using TensorFlow as a back-end
- You can work exclusively in the high-level Keras environment
- Both are in active development with large user bases so there is much help available online
- Both are easy to install using `pip`:

Notes on Frameworks

- As stated previosuly, for this assignment I would recommend Keras using TensorFlow as a back-end
- You can work exclusively in the high-level Keras environment
- Both are in active development with large user bases so there is much help available online
- Both are easy to install using `pip`:

```
$ pip install keras
$ pip install tensorflow-gpu
```

Note: for the CPU version use `pip install tensorflow`

Data Set

ISIC Archive

There are *some* comprehensive open medical data sets. One such data set is the ISIC Archive (The International Skin Imaging Collaboration)

- URL: `https://isic-archive.com/` provides an image browser, images downloaded via an API

ISIC Archive

There are *some* comprehensive open medical data sets. One such data set is the ISIC Archive (The International Skin Imaging Collaboration)

- URL: `https://isic-archive.com/` provides an image browser, images downloaded via an API
- Currently there are 13,791 high-resolution dermascopy images in the database

ISIC Archive

There are *some* comprehensive open medical data sets. One such data set is the ISIC Archive (The International Skin Imaging Collaboration)

- URL: `https://isic-archive.com/` provides an image browser, images downloaded via an API
- Currently there are 13,791 high-resolution dermascopy images in the database
- Released under the Creative Commons Licence

ISIC Archive

There are *some* comprehensive open medical data sets. One such data set is the ISIC Archive (The International Skin Imaging Collaboration)

- URL: `https://isic-archive.com/` provides an image browser, images downloaded via an API
- Currently there are 13,791 high-resolution dermascopy images in the database
- Released under the Creative Commons Licence
- This means: *You can copy, modify, distribute and perform the work, even for commercial purposes, all without asking permission.*[4]

[4] `http://creativecommons.org/publicdomain/zero/1.0/`

ISIC Archive

There are *some* comprehensive open medical data sets. One such data set is the ISIC Archive (The International Skin Imaging Collaboration)

- URL: `https://isic-archive.com/` provides an image browser, images downloaded via an API
- Currently there are 13,791 high-resolution dermascopy images in the database
- Released under the Creative Commons Licence
- This means: *You can copy, modify, distribute and perform the work, even for commercial purposes, all without asking permission.*[4]
- The **ISIC archive data set will be the basis of this assignment's data generation task**

[4] `http://creativecommons.org/publicdomain/zero/1.0/`

ISIC Archive

The ISIC data set is a skin lesion image data set.

[5] Credit
https://en.wikipedia.org/wiki/Dermatoscopy#/media/File:Dermatoscope1.JPG

ISIC Archive

The ISIC data set is a skin lesion image data set.

- The images are taken with a **dermatoscope**[5]

[5] Credit
https://en.wikipedia.org/wiki/Dermatoscopy#/media/File:Dermatoscope1.JPG

ISIC Archive

The ISIC data set is a skin lesion image data set.

- The images are taken with a **dermatoscope**[5]

[5] Credit
https://en.wikipedia.org/wiki/Dermatoscopy#/media/File:Dermatoscope1.JPG

ISIC Archive

The ISIC data set is a skin lesion image data set.

- The images are taken with a **dermatoscope**[5]

- This device records high resolution images taken at approx $10\times$ zoom

[5] Credit
https://en.wikipedia.org/wiki/Dermatoscopy#/media/File:Dermatoscope1.JPG

ISIC Archive

The ISIC data set is a skin lesion image data set.

- The images are taken with a **dermatoscope**[5]

- This device records high resolution images taken at approx 10× zoom
- The lesion is completely enclosed so there are is little reflected light from surroundings (resulting in more standardised images)

[5] Credit
https://en.wikipedia.org/wiki/Dermatoscopy#/media/File:Dermatoscope1.JPG

ISIC Archive

The ISIC data set is a skin lesion image data set.

- The images are taken with a **dermatoscope**[5]

- This device records high resolution images taken at approx $10\times$ zoom
- The lesion is completely enclosed so there are is little reflected light from surroundings (resulting in more standardised images)
- Non-invasive procedure

[5] Credit
https://en.wikipedia.org/wiki/Dermatoscopy#/media/File:Dermatoscope1.JPG

ISIC Archive

The ISIC archive concerns skin lesions and skin cancer.

ISIC Archive

The ISIC archive concerns skin lesions and skin cancer.

- Skin lesions are **either** malignant (dangerous, cancer, can metastasise) or benign

ISIC Archive

The ISIC archive concerns skin lesions and skin cancer.

- Skin lesions are **either** malignant (dangerous, cancer, can metastasise) or benign
- However, within these two types are many more **diagnoses** which are subcategories, such as the standard malignant melanoma or benign diagnoses such as basal cell papilloma or, of course, common nevi (birthmarks, moles)

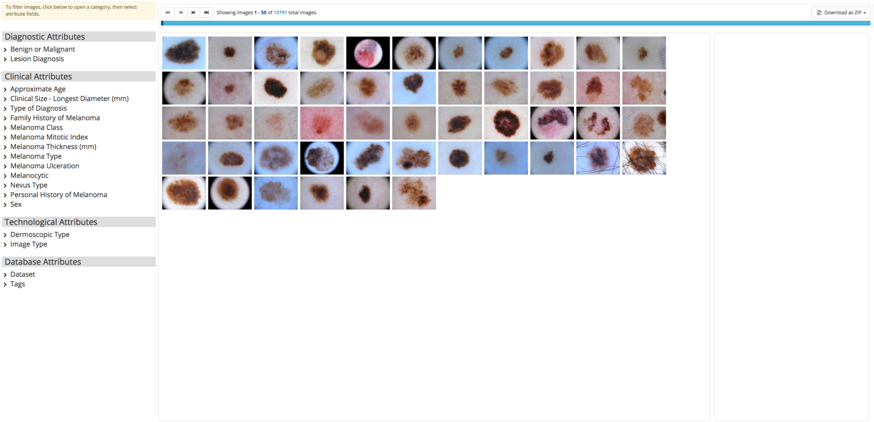
ISIC Archive

The ISIC archive concerns skin lesions and skin cancer.

- Skin lesions are **either** malignant (dangerous, cancer, can metastasise) or benign
- However, within these two types are many more **diagnoses** which are subcategories, such as the standard malignant melanoma or benign diagnoses such as basal cell papilloma or, of course, common nevi (birthmarks, moles)

For our work here, **we do not have to concern ourselves with whether the lesions are malignant or benign**. We are also **not concerned with their diagnosis**, we only want to generate realistic looking images. More on this later.

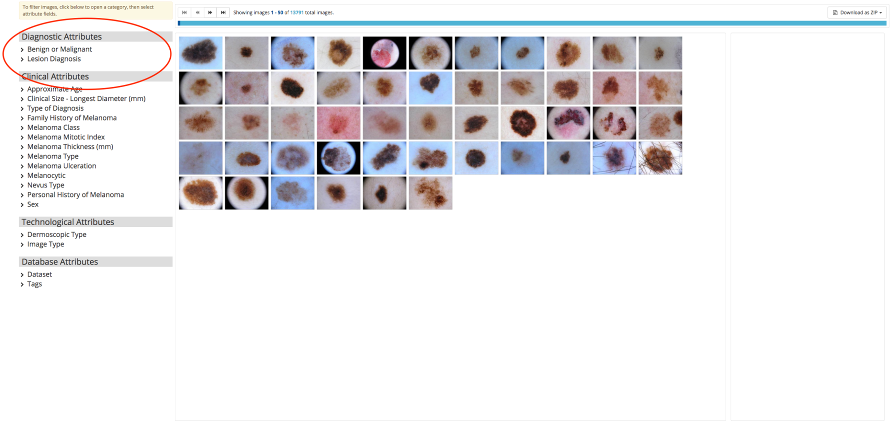
ISIC Archive



URL:

`http://display.isic-archive.com/#!/onlyHeaderTop/gallery`

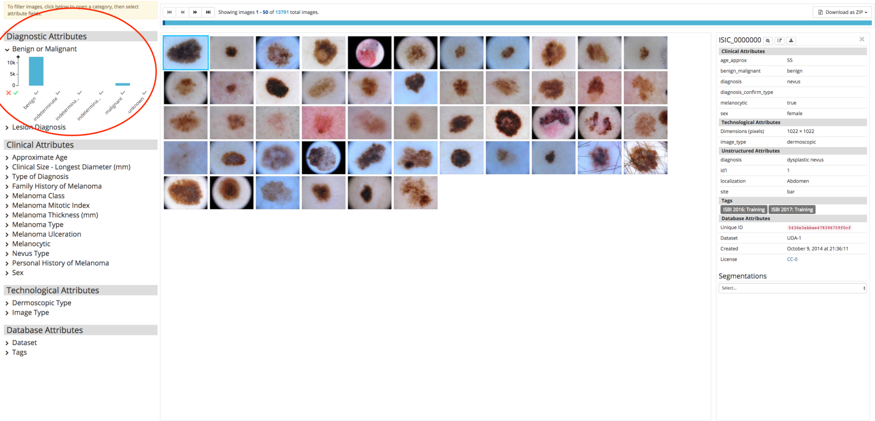
ISIC Archive



URL:

`http://display.isic-archive.com/#!/onlyHeaderTop/gallery`

ISIC Archive



URL:

`http://display.isic-archive.com/#!/onlyHeaderTop/gallery`

ISIC Archive: Benign Nevus

ISIC Archive: Malignant Melanoma

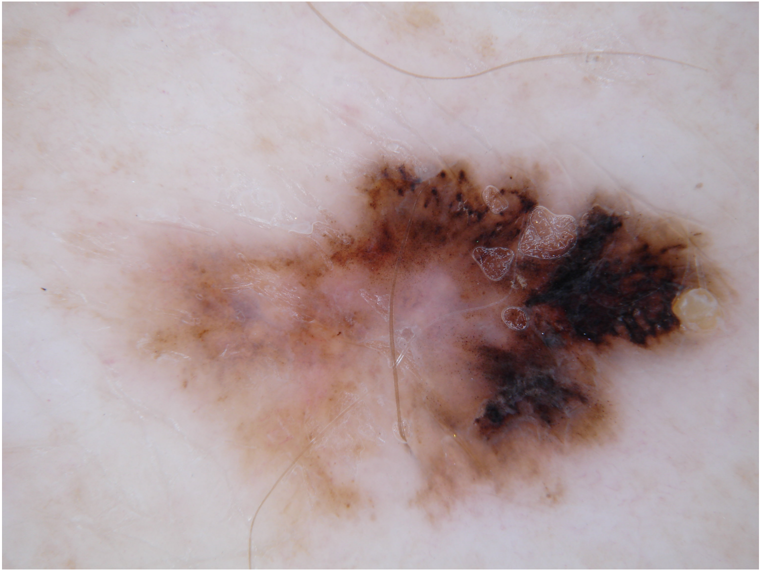


Image Details

Benign nevus image:

- Filename: ISIC_0000000.jpg
- Unique ID: 5436e3abbae478396759f0cf
- URL: `https://isic-archive.com/api/v1/image/5436e3abbae478396759f0cf/download`

Malignant melanoma image:

- Filename: ISIC_0000390.jpg
- Unique ID: 5436e3dcbae478396759f3dd
- URL: `https://isic-archive.com/api/v1/image/5436e3dcbae478396759f3dd/download`

ISIC Archive

Some **important** notes regarding the dataset:

- While there are 13,791 images there is a very large class-imbalance

ISIC Archive

Some **important** notes regarding the dataset:

- While there are 13,791 images there is a very large class-imbalance
- There are 12,668 benign images and only 1,084 malignant images

ISIC Archive

Some **important** notes regarding the dataset:

- While there are 13,791 images there is a very large class-imbalance
- There are 12,668 benign images and only 1,084 malignant images
- When creating a GAN to generate the data, you **might** want to select ***either* the benign or malignant samples**

ISIC Archive

Some **important** notes regarding the dataset:

- While there are 13,791 images there is a very large class-imbalance
- There are 12,668 benign images and only 1,084 malignant images
- When creating a GAN to generate the data, you **might** want to select ***either* the benign or malignant samples**
- This may result in more realistic looking images as we do not want create artificial image data that contains a mixture of the malignant and benign features

ISIC Archive

Some **important** notes regarding the dataset:

- While there are 13,791 images there is a very large class-imbalance
- There are 12,668 benign images and only 1,084 malignant images
- When creating a GAN to generate the data, you **might** want to select ***either* the benign or malignant samples**
- This may result in more realistic looking images as we do not want create artificial image data that contains a mixture of the malignant and benign features
- As seen in the previous slide, malignant and benign lesions *can* look quite different

ISIC Archive

Some **important** notes regarding the dataset:

- While there are 13,791 images there is a very large class-imbalance
- There are 12,668 benign images and only 1,084 malignant images
- When creating a GAN to generate the data, you **might** want to select ***either* the benign or malignant samples**
- This may result in more realistic looking images as we do not want create artificial image data that contains a mixture of the malignant and benign features
- As seen in the previous slide, malignant and benign lesions *can* look quite different
- However, some types of generative networks can handle data in seperate classes, it depends on the type of network you design

Assignment

Assignment

Goal: **generate synthetic medical image data as realistically as possible**

- Group work

Assignment

Goal: **generate synthetic medical image data as realistically as possible**

- Group work
- Use Generative Adversarial Networks of some kind

Assignment

Goal: **generate synthetic medical image data as realistically as possible**

- Group work
- Use Generative Adversarial Networks of some kind
- Ideally someone in the group has access to a GPU

Assignment

Goal: **generate synthetic medical image data as realistically as possible**

- Group work
- Use Generative Adversarial Networks of some kind
- Ideally someone in the group has access to a GPU
- Task is to **generate skin lesion images which are as realistic as possible** in order to see **if it is possible to convince a dermatologist that they are seeing real data**

Assignment

Goal: **generate synthetic medical image data as realistically as possible**

- Group work
- Use Generative Adversarial Networks of some kind
- Ideally someone in the group has access to a GPU
- Task is to **generate skin lesion images which are as realistic as possible** in order to see **if it is possible to convince a dermatologist that they are seeing real data**
- We will use the ISIC archive as our basis data set

Assignment

Goal: **generate synthetic medical image data as realistically as possible**

- Group work
- Use Generative Adversarial Networks of some kind
- Ideally someone in the group has access to a GPU
- Task is to **generate skin lesion images which are as realistic as possible** in order to see **if it is possible to convince a dermatologist that they are seeing real data**
- We will use the ISIC archive as our basis data set
- Each group will be evaluated by a dermatologist to see which group created the best fake data

Assignment

Goal: **generate synthetic medical image data as realistically as possible**

- Group work
- Use Generative Adversarial Networks of some kind
- Ideally someone in the group has access to a GPU
- Task is to **generate skin lesion images which are as realistic as possible** in order to see **if it is possible to convince a dermatologist that they are seeing real data**
- We will use the ISIC archive as our basis data set
- Each group will be evaluated by a dermatologist to see which group created the best fake data
- **Competition to see which group can generate the best fake images!**

Notes on Frameworks

There are a few requirements about the frameworks you choose and how you submit your work.

- For this assignment you **must use a Python-based framework**

Notes on Frameworks

There are a few requirements about the frameworks you choose and how you submit your work.

- For this assignment you **must use a Python-based framework**
- Most frameworks are either **Python-based or have Python bindings**

Notes on Frameworks

There are a few requirements about the frameworks you choose and how you submit your work.

- For this assignment you **must use a Python-based framework**
- Most frameworks are either **Python-based or have Python bindings**
- Your submissions should be **in the form of Jupyter Notebooks containing your source code and proper documentation and comments**

Notes on Frameworks

There are a few requirements about the frameworks you choose and how you submit your work.

- For this assignment you **must use a Python-based framework**
- Most frameworks are either **Python-based or have Python bindings**
- Your submissions should be **in the form of Jupyter Notebooks containing your source code and proper documentation and comments**

Jupyter is a web-based development environment, where you work with *notebooks* which can easily be shared

Notes on Frameworks

There are a few requirements about the frameworks you choose and how you submit your work.

- For this assignment you **must use a Python-based framework**
- Most frameworks are either **Python-based or have Python bindings**
- Your submissions should be **in the form of Jupyter Notebooks containing your source code and proper documentation and comments**

Jupyter is a web-based development environment, where you work with *notebooks* which can easily be shared

- Install it using `pip`:

Notes on Frameworks

There are a few requirements about the frameworks you choose and how you submit your work.

- For this assignment you **must use a Python-based framework**
- Most frameworks are either **Python-based or have Python bindings**
- Your submissions should be **in the form of Jupyter Notebooks containing your source code and proper documentation and comments**

Jupyter is a web-based development environment, where you work with *notebooks* which can easily be shared

- Install it using `pip`:

```
$ pip install jupyter
```

Notes on Frameworks

Augmentor_Keras_Array_Data - Mozilla Firefox

Augmentor_Keras_Ar ×

localhost:8888/notebooks/Augmentor_Keras_Array_Data.ipynb

Suchen

jupyter Augmentor_Keras_Array_Data (autosaved)

File Edit View Insert Cell Kernel Widgets Help

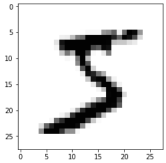
Python 2

Markdown CellToolbar

Out[5]: (28, 28)

We can use matplotlib's imshow function to render this array as an image:

In [6]:
```
plt.imshow(x_train[0], cmap="Greys");
```



Later, we will pass this entire matrix, containing 60,000 images, to an Augmentor function, which will generate batches of augmented images from this original data.

Create a Pipeline

It is perfectly fine to create a pipeline object without needing to point to a directory on your hard drive. Do this if you want to perform an augmentation task on data from another location, such as from the web or another framework such as Keras.

In [7]:
```
p = Augmentor.Pipeline()
```

Assignment

Some hints to get you started:

- All groups should use the ISIC archive as a data source, if you want you can use only the benign images

Assignment

Some hints to get you started:

- All groups should use the ISIC archive as a data source, if you want you can use only the benign images
- Only use both malignant and benign if your network is designed for this

Assignment

Some hints to get you started:

- All groups should use the ISIC archive as a data source, if you want you can use only the benign images
- Only use both malignant and benign if your network is designed for this
- If training is very long, then reduce the resolution of your input/output images

Assignment

Some hints to get you started:

- All groups should use the ISIC archive as a data source, if you want you can use only the benign images
- Only use both malignant and benign if your network is designed for this
- If training is very long, then reduce the resolution of your input/output images
- A GPU will massively speed up training so ideally each group should have access to a GPU (you can of course test code on a CPU)

Assignment

Some hints to get you started:

- All groups should use the ISIC archive as a data source, if you want you can use only the benign images
- Only use both malignant and benign if your network is designed for this
- If training is very long, then reduce the resolution of your input/output images
- A GPU will massively speed up training so ideally each group should have access to a GPU (you can of course test code on a CPU)
- It might be useful if each group had one person who is familiar with Python

Assignment

Some hints to get you started:

- All groups should use the ISIC archive as a data source, if you want you can use only the benign images
- Only use both malignant and benign if your network is designed for this
- If training is very long, then reduce the resolution of your input/output images
- A GPU will massively speed up training so ideally each group should have access to a GPU (you can of course test code on a CPU)
- It might be useful if each group had one person who is familiar with Python

Assignment

Some hints to get you started:

- All groups should use the ISIC archive as a data source, if you want you can use only the benign images
- Only use both malignant and benign if your network is designed for this
- If training is very long, then reduce the resolution of your input/output images
- A GPU will massively speed up training so ideally each group should have access to a GPU (you can of course test code on a CPU)
- It might be useful if each group had one person who is familiar with Python

A good first step might be to read Salimans et al. on improved GANs `https://arxiv.org/abs/1606.03498` and the corresponding GitHub repository: `https://github.com/openai/improved-gan` or the work on DCGANs by Radford et al. `https://arxiv.org/abs/1511.06434`

Questions

Any Questions?

Throughout the semester you can visit this Gitter channel to ask questions:

https://gitter.im/MLHI

Or email me:

`marcus.bloice@medunigraz.at`

References and Links

Here are a few papers and links you might want to check out:

- Guibas, Virdi, Li, **Synthetic Medical Images from Dual Generative Adversarial Networks**, *arXiv*, 2018: `https://arxiv.org/abs/1709.01872`
- Tutorial on GANs with source code for GAN in Python/Keras: `https://deeplearning4j.org/generative-adversarial-network`
- Implementing a GAN with Keras and a TensorFlow back-end: `https://towardsdatascience.com/gan-by-example-using-keras-on-tensorflow-backend-1a6d515a`
- The `Keras-Adversarial` project: `https://github.com/bstriner/keras-adversarial`
- Ian Goodfellow's tutorial at NIPS 2016: `http://www.iangoodfellow.com/slides/2016-12-04-NIPS.pdf`

References and Links

Here are a few papers and links you might want to check out:

- Curated list of GAN advancements: `http://gkalliatakis.com/blog/delving-deep-into-gans`
- DCGAN with source: `https://carpedm20.github.io/faces/`
- Improved GANs paper description: `https://towardsdatascience.com/semi-supervised-learning-with-gans-9f3cb128c5e`
- Good overview article: `http://gkalliatakis.com/blog/delving-deep-into-gans`

References and Links

Some notes regarding semi-supervised and unsupervised approaches:

- See Salimans, 2016: Improved Techniques for Training GANs. This paper address semi-supervised approaches. Salimans, Tim, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. **Improved techniques for training gans**. In *Advances in Neural Information Processing Systems*, pp. 2234-2242. 2016. See also: `https://github.com/openai/improved-gan`.
- Springenberg, J. **Unsupervised and Semi-supervised Learning with Categorical Generative Adversarial Networks**. *arXiv preprint arXiv:1511.06390* (2015).