Joonas Tuomikoski

Unsupervised feature analysis of real and synthetic knee X-ray images

Master's Thesis in Information Technology

June 14, 2023

University of Jyväskylä

Faculty of Information Technology

Author: Joonas Tuomikoski

Contact information: joonas.a.s.tuomikoski@jyu.fi

Supervisors: Sami Äyrämö, and Fabi Prezja

Title: Unsupervised feature analysis of real and synthetic knee X-ray images

Työn nimi: Ohjaamaton piirreanalyysi aidoille ja synteettisille röntgenkuville

Project: Master's Thesis

Study line: Laskennalliset tieteet

Page count: 46+0

Abstract: Generative models have improved massively in the recent years, and this has created a need for automatic validation techniques for synthetic data. In this master's thesis a method for validating synthetic images based on feature extraction and cluster analysis is tested on X-ray images created with generative adversarial networks. The results show that the generated images follow the distribution of the imageset used in training, but are clearly distinct from a different X-ray imageset.

Keywords: cluster analysis, convolutional neural networks generative models, synthetic data, osteoarthritis, validation

Suomenkielinen tiivistelmä: Generatiiviset mallit ovat parantuneet valtavasti viime vuosina, ja tämä on luonut tarpeen automaattisille validointitekniikoille synteettiselle datalle. Tässä pro gradu -työssä testatiin menetelmää synteettisten kuvien validointiin, joka perustuu piirteiden poimimiseen ja klusterianalyysiin, generatiivisten vastakkaisten verkostojen luotujen röntgenkuvien avulla. Tulokset osoittavat, että luodut kuvat noudattavat koulutuksessa käytettyjen kuvien jakaumaa, mutta eroavat selvästi toisesta datajoukosta olevista röntgenkuvista.

Avainsanat: klusterianalyysi, konvoluutioneuroverkot, generatiiviset mallit, synteettinen data, nivelrikko, validointi

List of Figures

List of Tables

Contents

1 Introduction

The field of deep learning has been growing rapidly in the last decade. However, developing new methods and training the neural networks used in deep learning requires a lot of data. This poses a large problem as healthcare information such as medical records and imaging results are highly sensitive, and cannot be freely shared as such. Large healthcare facilities and research centers have access to their own data, but smaller companies and independent researchers might have obstacles accessing such data. Combined datasets could also help the models generalize more.

Medical data is protected with the restrictions of Health Insurance Portability Accountability Act HIPAA (Health and Services 2000) and General Data Protection Regulation GDPR (European Parliament 2016). While these require only de-identification (i.e. removal of identifiable information and anonymization), it has proven to be a weak measure to protect the identity of the people present in the dataset (Narayanan and Felten 2014). One solution to this is to synthesize new data, and then share the newly created data for further analysis, without endangering the privacy of the individuals in the original database.

Testing this newly created synthetic data and validating it is important. The synthetic data should represent the original data as well as possible while simultaneously being different enough to achieve privacy. Validation can be done using expert knowledge, for example by having them evaluate the "realness" of synthetic X-ray images. This is very time-consuming, and therefore a need for automatic validation methods exists.

The goal of this Master's thesis is to explore a method for validating synthetic X-ray images. In the following chapter 2 the problem domain, knee osteoarthritis, is introduced. Then in chapter 3 the basics of convolutional neural networks and transfer learning are gone over, leading to generative adversarial networks and cluster analysis in chapters 4 and 5, which concludes the theoretical part. Then the datasets used, the methods, and the execution of the experiment are detailed in chapter 6. In chapter 7 the results of the experiment are shown, and chapter 8 concludes this thesis with discussion the results, their impact, and how the research could be built upon in the future.

2 Knee osteoarthritis

Osteoarthritis is the most common joint disease in the world (Käypä hoito 2016). It is a degenerative disease which cannot be cured, and causes pain, stiffness inactivity, and eventually loss of ability to function (Wieland et al. 2005). In addition to the decrease in quality of life for the individual, it causes major socioeconomic burdens with care costs and productivity loss (Hunter, Schofield, and Callander 2014).

Osteoarthritis is the most prevalent in the knee joint: In the Health 2000 -survey in Finland the prevalence of clinically diagnosed knee osteoarthritis was found to be 6.1% in men and 8.0% in women (age-adjusted) (Kaila-Kangas 2007). The prevalence rises steeply with age, and in the age group of over 85 44.2% of men and 35.6% of women were diagnosed with knee osteoarthritis. The strongest predictors other than age for osteoarthritis are obesity and strenuous physical activity, such as working in a physically demanding occupation (Lespasio et al. 2017).

The treatment of knee osteoarthritis aims for pain mitigation and reduction, upholding and increasing functional ability, and halting the progression of the disease (Käypä hoito 2016). The treatment options include weight loss, exercise therapy, pain medication, assistive devices such as shock-absorbing shoes, and surgical procedures (Lespasio et al. 2017). Because preventative measures (e.g. weight monitoring, exercise) are preferred, early detection is paramount (Roos and Arden 2016).

2.1 Kellgren-Lawrence -grading

Osteoarthritis is commonly diagnosed with clinical assessment and confirmed using radiographic imaging. Normal X-ray imaging is usually enough, and MRI or tomographic imaging is rarely needed (Sinusas 2012). Knee osteoarthritis manifests in X-rays mainly with formation of osteophytes (small bone formations in the joint area), and joint-space narrowing (Sinusas 2012)

Kellgren-Lawrence -grading (Kellgren and Lawrence 1957) is one of the most used grading

systems to assess the severity of osteoarthritis from radiographs. It has five different levels: None (0), Doubtful (1), Minimal (2), Moderate (3), and Severe (4). An example image from each of these levels of osteoarthritis are shown in figure 1. While discerning between grades 0, 1 and 2 is greatly beneficial in terms of early intervention, the differences can be very subtle.

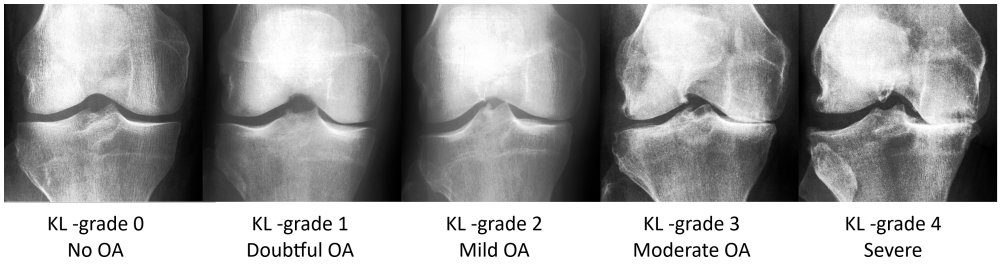

Figure 1. Examples of Kellgren-Lawrence (KL) grading system by presence/severity of osteoarthritis (OA)

2.2 Previous research

Kellgren-Lawrence -grades have been widely used in machine learning applications (Lee et al. 2022). However, as was noted in the original paper by Kellgren and Lawrence, there is a notable variance in medical experts' grades, with both intraobserver and interobserver correlation being 0.83 in grading knee osteoarthritis severity (Kellgren and Lawrence 1957). This makes training a grade predicting model especially difficult: As an example, a model by Tiulpin et. al. achieved a multiclass accuracy of 66.71%, while the agreement with experts on the dataset was 83% (Tiulpin et al. 2018).

In the last few years multiple data synthetization methods have been researched using health-care data (Murtaza et al. 2023). One notable method and research in this field was medGAN (Choi et al. 2017), which generated discrete patient records using Generative Adversarial Networks (see chapter 4). Data synthetization methods have also been utilized for medical image generation. For example, in 2021 Karbhari et al. generated chest X-rays of COVID-19 -patients, and showed that the accuracy of classification models either improved or remained the same when augmenting the training data with synthetic data (Karbhari et al. May 2021).

3 Convolutional neural networks

Convolutional neural networks (CNN) are based on the convolution operation and have become the most widely deep learning structure for computer vision. As well as other neural networks, CNNs have a long history from the first perceptron (Rosenblatt 1958) to the neurocogitron (Fukushima 1980), before arriving to modern CNNs (e.g. AlexNet (Krizhevsky, Sutskever, and Hinton 2017)).

In the following sections the driving force behind CNNs, the convolution operation, is described, before moving on to the basic structure of CNNs. Then the concept of using images from another domain to learn general features of images together with tools to assist in that endeavor are explored. Finally a method of visualizing essential parts of an image is presented with class-activation mapping.

3.1 Convolution

Convolution is a mathematical operation between two functions, defined by the following integral:

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau \tag{3.1}$$

Convolution appears in this form for example in probability theory, as the convolution of two (independent) probability density functions P_X and P_Y is the sum of those two functions, P_{X+Y}. But as images often consist of discrete pixels, a discretized convolution can be used:

$$(f * g)(n) = \sum_{m=-\infty}^{\infty} f(m)g(n-m) \tag{3.2}$$

When applied to images the first function f can be thought of as a representation of an image, and g as a *kernel*. The convolution then applies some kernel-dependent transformation to each pixel of the image, and the image is transformed when the kernel is moved through every pixel of the image. An example of a edge detecting kernel applied to an image can be seen in figure 2. As the kernel is often not mirrored in x and y directions, the convolution in

computer vision is most of the time actually *cross-correlation*. It is still traditionally called convolution, as the mirroring has no practical effect on CNNs. In the simple case of black and white images this 2-d convolution operation takes the form

$$(f * g)(i, j) = \sum_m \sum_n f(i+m, j+n) g(m, n), \tag{3.3}$$

where f and g are the image and kernel respectively, i and j are the coordinates of the pixel, and m and n are the dimensions of the kernel.

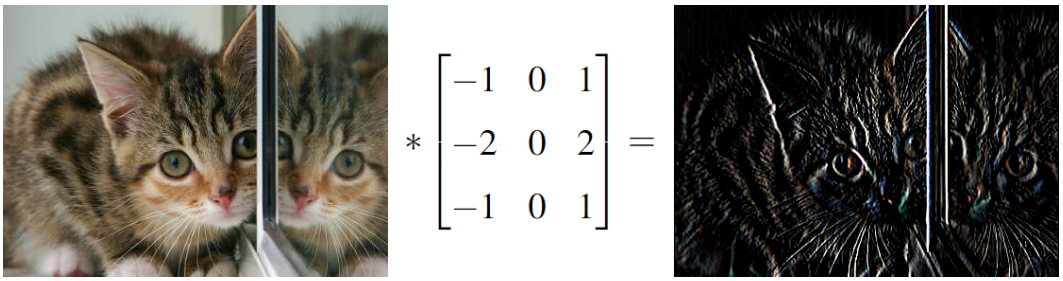

Figure 2. A picture convolved with an edge detecting kernel

Manually chosen kernels, like the edge detection kernel, can be useful in tasks like finding regions of interest in images, for example finding faces in images (Viola and Jones 2004) or bone segmentation (Lindner et al. 2013). The major advantage of these simple manually chosen kernels is their and that they can be used with other machine learning techniques such as random forests. Their disadvantage is that manually crafting kernels is difficult. While horizontal and vertical edge detectors are intuitive, images consist of more complex features. By using deep convolutional networks and backpropagation these features can be learnt from the data.

3.2 Deep convolutional neural networks

Convolutional neural networks follow the same structure as "normal", fully-connected multilayer perceptrons; They have an input layer, multiple hidden layers, and an output layer. In a regular image classification task the input layer takes the image as a pixel intensity matrix (with values often normalized to $[0, 1]$), and the output is, for example, the probability of the image containing a cat or probabilities of multiple classes. In theory one could use fully

connected layers in hidden layers and have a functioning classifier, but in practice this has several disadvantages.

The amount of parameters explodes rapidly with image size, as every connection adds one parameter (and every node has a bias). The number of parameters is given by $\sum_{n=1}^{m} n_{l-1} n_l + n_l$, where n_l is the number of nodes in a layer l and m is the number of subsequent layers after the input layer (layer 0). It can be easily seen that if every pixel value of an image is used as the input the number of parameters quickly rises to billions even when using moderately deep network architectures and fairly small images. The fully connectedness also means that, in terms of network architecture, every pixel contributes equally to the output. This leads to, for example, uniform white pixels in the background affecting the output as much as the subject of the image. While the background of an object can be quite useful in identifying tasks, giving every pixel equal importance makes the learning unnecessarily hard. Neighboring pixels having no special connection also means that there is no locality.

CNNs address these problems using convolution and pooling layers. A convolution layer consists of n kernels, and the parameters of these kernels are learned with backpropagation. Because the kernels are usually smaller than the image, this means the number of parameters is reduced, as the same operation is performed in all parts of the image. A non-linear activation function is used after the convolution, similarly to a fully-connected neural network, and the outputs of the different kernels result in n image-like feature-maps which are propagated forwards. The convolution layers have been described to focus on larger concepts on each layer: First layer detecting lines, the second layer detecting corners and other constructs formed in this line-feature-map, and so on until the final layer and objects can be detected.

The convolution and activation is usually followed by a pooling layer, such as a max-pooling layer (Goodfellow, Bengio, and Courville 2016). Max-pooling takes the maximum value of a $n \times m$ grid on every position of a feature-map, and generates a new feature-map of these maximum values. This has two effects: Firstly, it decreases the dimensions of the feature-map, especially when the max-pooling grip is moved across the feature-map in strides larger than 1. This frees computation to be used, for example, for more convolution kernels. Secondly, it add a slight invariance to the feature-map. It does not matter in which pixel in the $n \times m$ grid the largest response is; the result stays the same even when the input is shifted slightly.

Other pooling layers exist, e.g. average-pooling which works similarly to max-pooling but takes the average of the grid instead of the maximum.

A simple CNN consists of convolution layers with varying kernel sizes alternating with pooling layers with varying grid sizes and stride lengths, ending with a few fully connected layers and a output layer. Examples of these include one of the first CNNs, LeNet (LeCun et al. 1998), and the CNN used in this research, VGG19 (Simonyan and Zisserman 2014), VGG19, among others, have pre-trained implementations freely available to use and modify, and this is discussed further in the following section.

3.3 Transfer learning, Tensorflow, and VGG19

Training a large CNN is computationally expensive and requires a large dataset. If the first convolution layers seemingly focus on more abstract features such as lines, it would be logical to reuse these layers and not to train them from the beginning of every new task, especially when there is only a small dataset available for the training. The usage of knowledge gained on a domain and applying it to a different domain is called transfer learning (Goodfellow, Bengio, and Courville 2016). In the case of computer vision this means training a CNN on a set of images, and applying it to a different set, with re-training or *fine-tuning* the last layers.

The approach of using pre-trained networks and fine-tuning them for a different domain is largely prevalent in medical image tasks, as medical images are highly sensitive and not easily available (Litjens et al. December 2017). This can be useful not only in transferring knowledge between similar tasks, such as from mammograms to digital breast tomography (Samala et al. 2016), but also from seemingly unrelated, natural images (i.e. non-medical images) to various fields in medicine (Shin et al. May 2016).

ImageNet (Deng et al. 2009) is an image dataset first introduced in 2009 and which currently has over 1.4 million natural images. It is largely used for pre-training, and has been shown advantageous when the training set in the application domain is small, but the advantages diminish with larger training set (He, Girshick, and Dollár 2019). One of the other advantages of CNNs pre-trained on ImageNet is their ease of availability. TensorFlow (Abadi et al. 2015), the framework used for neural networks in this research, allows the use of the

high level API Keras, which houses a selection of implementations of some of the more well-known CNN models, including the model used in this research, VGG19 (Simonyan and Zisserman 2014). Moreover, these models can be loaded with weights gained from pre-training with ImageNet.

VGG19 (Simonyan and Zisserman 2014) is a CNN which was developed by Visual Geometry Group, University of Oxford, and has 19 layers (not counting max-pooling layers). The architecture of VGG19 is fairly traditional; it consists of five convolution blocks each followed by a pooling layer, ending with three fully connected layers. A detailed architecture is seen in figure 3. According to the Keras API website[1] it has accuracy of 71.3% and 90.0% on ImageNet test set on the first prediction being correct and one of the top-5 predictions being correct respectively. This is not the top performance of the models included in Keras, but VGG19 has a straightforward, explainable architecture which is useful in in decoding which parts of the image lead to a prediction.

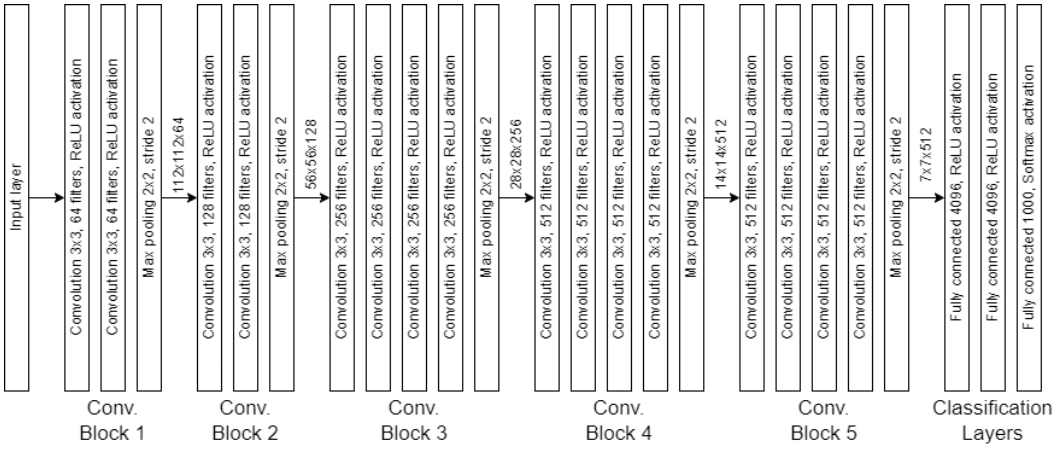

Figure 3. The architecture of the VGG19 convolutional neural network

3.4 Class-activation mapping

Neural networks can be thought of as "black box systems": The input and the output are easy to understand, but the hidden layers with millions of parameters are indecipherable by humans. This poses a problem: How can a model be trusted if we can only observe the inputs

1. https://keras.io/api/applications/

and outputs?

This is the research field of Explainable AI, which attempts to provide evidence or a reason for an AI system's outputs (Phillips et al. 2020). Class-activation mapping (CAM) (Zhou et al. 2016) is a method used in attempt to achieve this in CNNs with visual explanation maps. The idea is simple: The CNN is given an image as an input and is propagated through the network, and the layers of the CNN are studied to map which parts of the image contributed to the labeling of the image into a certain class.

There are many different methods based on CAM (Jung and Oh 2021). One of the earliest is Grad-CAM (Selvaraju et al. 2017), which takes the gradients with regards to the class activation

4 Generative Adversarial Networks

Generative adversarial networks (GANs) are a framework (see figure 4) to train generative models with an adversarial process (Goodfellow et al. 2014). The original idea for GANs was to train simultaneously two competing models, a generator G and a discriminator D. The generator G would attempt to model the training data distribution and synthesize new samples from that distribution, while the discriminator D would try to determine if the sample was from the real training data or a new synthesized sample.

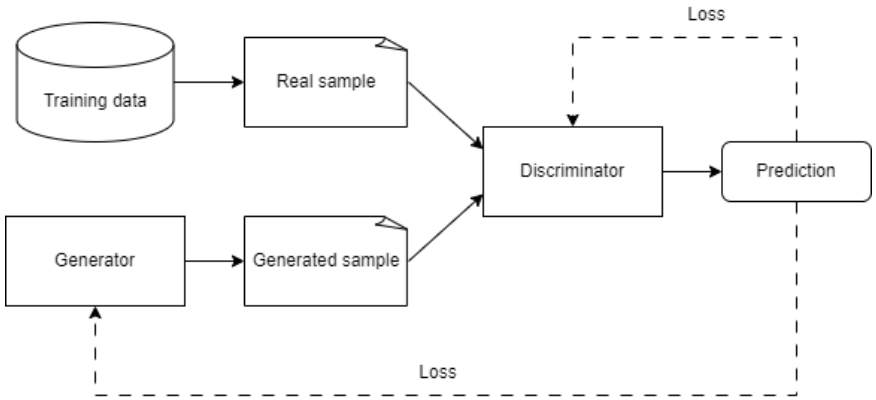

Figure 4. Architecture of the GAN framework

Both of the models would be trained using backpropagation using the value function $V(D,G)$ shown in equation 4.1. As this is a two-player minimax game where when the other succeed the other fails, the desired equilibrium is attained when the discriminator D estimates the probability of the sample being from the training data to be $P = 0.5$ for every sample. In this equilibrium the generator G has sufficiently captured the data distribution.

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim\mathbb{P}_r}[logD(x)] + \mathbb{E}_{z\sim\mathbb{P}_g}[log(1-D(x))] \tag{4.1}$$

While in theory this minimax-game seems like a valid strategy, in practice training a GAN is a lot harder and the convergence to an optimum is not guaranteed (Goodfellow, Bengio, and Courville 2016). Especially the generator gradient tends to vanish when the discriminator is very confident (Arjovsky and Bottou 2017). This has resulted in many improvements on the original framework, such as the Wasserstein GAN (Arjovsky, Chintala, and Bottou 2017).

Wasserstein GAN (WGAN) modifies the original value function 4.1 by making use of the Earth-Mover or Wasserstein-1 distance

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma}[\|x-y\|]$$

where $\Pi(\mathbb{P}_r, \mathbb{P}_g)$ is the set of all joint distributions $\gamma(x,y)$ whose marginals are respectively $\mathbb{P}_r$ and $\mathbb{P}_g$ (Arjovsky, Chintala, and Bottou 2017). Using Kantorovich-Rubinstein duality the problem is transformed into (Villani et al. 2009)

$$\max_{w \in W} \mathbb{E}_{x\sim\mathbb{P}_r}[f_w(x)] - \mathbb{E}_{z\sim p_{p(z)}}[f_w(g_\theta(z))]$$

which is similar to the original GAN-game with the notable difference of the discriminator being replaced with critic f_w with a linear output. f_w also needs to be 1-Lipschitz continuous, and this is achieved in WGAN by constraining the weights w to a range $[-c, c]$ by clipping.

WGANs were further improved upon with WGAN-GP using gradient penalty (Gulrajani et al. 2017). In WGAN-GP the 1-Lipschitz continuity is enforced by constraining the critic's gradient norm of the output with respect to the input:

$$L = \mathbb{E}_{x\sim\mathbb{P}_g}[f_w(x)] - \mathbb{E}_{x\sim\mathbb{P}_r}[f_w(x)] + \lambda \mathbb{E}_{\hat{x}\sim\mathbb{P}_{\hat{x}}}(\|\nabla_{\hat{x}} f_w(\hat{x})\|_2 - 1)^2], \tag{4.2}$$

where $\mathbb{P}_{\hat{x}}$ is defined as uniformly sampled between points sampled from $\mathbb{P}_r$ and $\mathbb{P}_g$.

4.1 FakeKnee-model

The synthetic images studied in this research were generated using convolutional WGAN-GP (Prezja et al. 2022). Two architecturally identical models were trained to generate X-rays of healthy knees (Kellgren-Lawrence grade 0 or 1) and knees with osteoarthritis (KL-grade 2, 3 or 4). The input to the generator was a noise vector of length 50 sampled from a standard normal distribution, which then used to generate $210 \times 210 \times 1$ -images using upsampling layers and convolution layers with activation layers using exponential linear units (ELU, Clevert, Unterthiner, and Hochreiter 2015). The discriminator (or critic) was a convolutional neural network with five convolution layers with ELU-activations, and dropout layers were used as regularizers. Finally, one linear dense layer was used as an output to calculate the gradients using 4.2 with gradient penalty $\lambda = 10$.

The generated images were validated using 15 medical experts. They were asked to identify whether a randomly sampled image from either the real set or the generated set was real or not. In total 60 images were used, and $61.35\%(\pm 10.71\%)$ total accuracy was achieved. The generated images were declared to have sufficient realism to deceive medical experts (Prezja et al. 2022). This result can be further tested using cluster analysis.

5 Cluster analysis

Cluster analysis is a form of *unsupervised learning*, as the data consists of vectors with no corresponding values and the aim is to discover samples which have features in common (Bishop and Nasrabadi 2006). The idea of transfer learning (see chapter 3.3) posits that images, for example, encoded as feature vectors could be clustered as the feature vectors of images which look alike should be similar. This allows extracting information from large unlabeled datasets, or when labels are difficult or even impossible to determine.

What is considered a cluster is not clearly defined. One abstract definition given classically has three parts: The members of a cluster should be as similar among themselves, they should be dissimilar to non-members of the cluster, and the measurement of similarity should be practical and explainable (Jain and Dubes 1988).

This three-part definition of a cluster allows many different clustering methods, two of which were used in this research. However when VGG19 is used to extract features from an image a vector of 512 values is produced. This is a very high dimensionality especially while considering the size of the data set (a total of 12000 images), and dimension reduction techniques should be considered prior to the actual clustering.

5.1 Dimension reduction

The *curse of dimensionality* refers to a set of problems which arise with larger dimensionality (Bishop and Nasrabadi 2006). For example if the similarity metric used in clustering is euclidean distance, it is easy to see that the data sparsity rises with the dimension count. This leads to overfitting, which in the case of clustering means that the number of possible clusters rises as the data points drift further apart. Also, in the case of image feature extraction, some features might be irrelevant for the image dataset. Below, two dimension reduction techniques used in this research are detailed.

5.1.1 Principal component analysis

The intuition behind principal component analysis (PCA) is to order the components (i.e. attributes or features) according to their variance. To reduce the dimensionality as much as possible while preserving most of the variation in the dataset, the components which contribute the least variance are discarded (Joliffe 2002).

Finding the principal components can be formulated as a problem of maximizing the variance of a data projected onto a lower dimensional plane. This problem simplifies to finding the eigenvectors of the data (Bishop and Nasrabadi 2006), and the principal components can be extracted with the following steps: Let $\mathbf{X}$ be a $N \times D$ matrix centered by subtracting the mean of each attribute. The covariance matrix can then be written as $\mathbf{C} = \mathbf{X}^T\mathbf{X}/(N-1)$, which is symmetric and can be diagonalized as $\mathbf{C} = \mathbf{V}\Lambda\mathbf{V}^{-1}$, where the column vectors of $\mathbf{V}$ are the eigenvectors of $\mathbf{C}$ and the diagonal values of Λ are the corresponding eigenvalues. The eigenvectors $\mathbf{V}$ are the principal axes and projecting the data $\mathbf{X}$ gives the principal components as the columns of $\mathbf{XV}$.

Singular value decomposition (SVD) is used in the method which was used used in this research to find the principal components[1]. With SVD the decomposition $\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^T$ is obtained, where $\mathbf{U}$ and $\mathbf{V}^T$ are unitary matrices and Σ is a singular matrix. The earlier equation for the covariance matrix $\mathbf{C} = \mathbf{X^T X}/(N-1)$ becomes

$$\mathbf{C} = \mathbf{V}\mathbf{S}\mathbf{U}^T\mathbf{U}\mathbf{S}\mathbf{V}^T/(N-1) = \mathbf{V}\frac{\mathbf{S}^2}{N-1}\mathbf{V}^T \tag{5.1}$$

as $\mathbf{U}$ was unitary. This shows that the vectors of $\mathbf{V}$ are the eigenvectors and the eigenvalues are given with $\lambda_i = \frac{s_i^2}{N-1}$. Thus the principal components are given by $\mathbf{XV} = \mathbf{U}\Sigma\mathbf{V}^T\mathbf{V} = \mathbf{U}\Sigma$. Because PCA is a linear operation, the projection onto a lower dimension can be represented as a layer of a neural network with a linear activation function.

5.1.2 UMAP

Uniform Manifold Approximation and Projection (UMAP) is a non-linear dimensionality reduction technique (McInnes, Healy, and Melville 2020). It has been used for visualization

1. https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html

purposes (e.g. Diaz-Papkovich et al. 2019) and for general dimensionality reduction purposes (e.g. Becht et al. 2019). It is similar to t-SNE (Van der Maaten and Hinton 2008) as both are based on manifold learning.

The overall goal of the UMAP algorithm is to learn a lower dimensionality embedding for the data where structures found in the original dimensionality are preserved. This means that neighboring datapoints in the original data should be close to each other in the embedding space, and non-neighboring points should remain distant.

First, a weighted k-neighbor graph is constructed. The k nearest neighbors for each datapoint x_i are searched using nearest neighbour descent (Dong, Moses, and Li 2011). The distance to the nearest neighbor of x_i is defined with distance metric d as

$$\rho_i = min\{d(x_i,\ x_{i_j}) \mid 1 \leq j \leq k,\ d(x_i,\ x_{i_j}) > 0\}$$

The weight function $w((x_i,\ x_{i_j}))$ is normalized with factor σ_i which is set to so that

$$\sum_{j=1}^{k} exp\left(\frac{-max(0,\ d(x_i,x_{i_j}) - \rho_i}{\sigma_i}\right) = log_2(k)$$

The weight function is then defined as

$$w((x_i,\ x_{i_j})) = exp\left(\frac{-max(0,\ d(x_i,x_{i_j}) - \rho_i}{\sigma_i}\right)$$

The weighted directed graph $\overline{G} = (V,E,w)$ can then be defined with vertices V being the set of datapoints, the edges $E = \{(x_i,x_{i_j}) | 1 \leq j \leq k, 1 \leq i \leq N\}$ with weights w. Because the directed edges from x_i to x_j can differ from x_j to x_i, the edges B of the graph G are obtained from

$$B = A + A^T - A \odot A^T$$

where A is the weighted adjacency matrix of $\overline{G}$ and $\odot$ is the Hadamard product.

The graph is first initialized using spectral embedding. It is then optimized by minimization

of fuzzy set cross entropy:

$$\begin{aligned} C((A,\mu),(A,\nu)) &= \sum_{a\in A}\left(\mu(a)log\left(\frac{\mu(a)}{\nu(a)}\right)+(1-\mu(a))log\left(\frac{1-\mu(a)}{1-\nu(a)}\right)\right) \\ &= \sum_{a\in A}(\mu(a)log(\mu(a))+(1-\mu(a))log(1-\mu(a))) \\ &\quad -\sum_{a\in A}(\mu(a)log(\nu(a))+(1-\mu(a))log(1-\nu(a))) \end{aligned}$$

where μ and ν are membership functions. Function μ corresponds to the edge strengths from B and ν is a similar membership function in the embedding dimension. As the first sum of the cross entropy depends only on μ the cost function to minimize is

$$C = -\sum_{a\in A}(\mu(a)log(\nu(a))+(1-\mu(a))log(1-\nu(a)))$$

Because μ is known membership function defined by B, a differentiable approximation for ν us needed to use stochastic descent on C:
Between two points $\mathbf{x}$ and $\mathbf{y}$ in lower dimensionality $\mathbb{R}^d$ the membership function is defined as

$$\Phi(\mathbf{x},\mathbf{y}) = \left(1+a(||\mathbf{x},\mathbf{y}||_2^2)^b\right)^{-1},$$

where parameters a and b are chosen by fitting a non-linear least squares against

$$\Psi(\mathbf{x},\mathbf{y}) = \begin{cases} 1 & \text{if } ||\mathbf{x},\mathbf{y}||_2 \leq min\_dist \\ \exp(-||\mathbf{x},\mathbf{y}||_2 - min\_dist) & otherwise \end{cases}$$

Hyperparameter *min_dist* is chosen to be the minimum distance between two points that are considered neighbors. Stochastic gradient descent is then performed for the number of epochs *n-epochs* w.r.t the gradient of the loss multiplied by learning rate

$$\alpha = 1 - epoch/n\text{-}epochs$$

.

While PCA is easily understandable intuitively and the embedding dimensions have a clear explanation as the orthogonal directions of maximal variance, a lot of topological information can be lost with dimension reduction. This is in contrast with UMAP, which attempts to

preserve the structures in the data, but the embedding dimensions do not have a clear meaning. The non-linear transformation of data can also lead to artificial clusters in clustering, as the datapoints moved according to possibly noisy data. For these reasons UMAP is used in tandem with PCA in this research, as an attempt to mitigate their respective drawbacks.

5.2 Clustering algorithms

The ambiguity in what a cluster is leads to many different ways to determine which datapoints constitute a cluster. For example, a cluster can be defined using some distance metric and and segmenting the data into groups. It can also be defined by searching concentrations of data with some density thresholds, or even by finding sets which locally follow some sort of hierarchical structure (Landau et al. 2011). Graphical examples of these scenarios can be seen in figure 5.

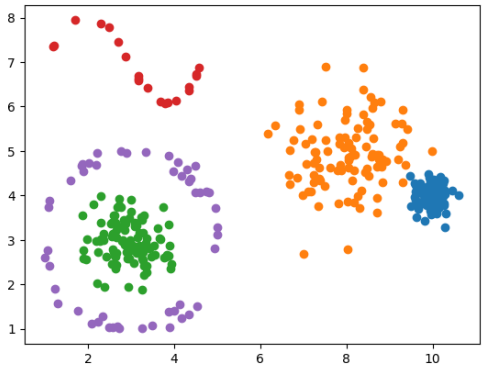

Figure 5. Examples of the multiple different ways a cluster can be defined

Finding clusters varying in sizes and shapes has given rise to multiple different techniques; one algorithm would struggle to find all the structures seen in figure 5. The two clustering algorithms, k-means and OPTICS, used in this research were chosen for similar reasons as to why PCA and UMAP were chosen as the dimensionality reduction, that is to have an easily explainable and controlled results of k-means, and the density based, more explorative qualities of OPTICS.

5.2.1 K-means

K-means, sometimes called the Lloyd's algorithm (Lloyd 1982), is a centroid-based clustering algorithm. It attempts to find an optimal clustering by assigning the datapoints into k groups and re-assigning the memberships to minimize the distance between datapoints and the mean of their respective groups. The clustering itself is not necessarily the global optimum as the k-means problem is NP-hard (Drineas et al. 2004). The standard k-means algorithm is shown in Algorithm 1.

Algorithm 1 K-means

$k \leftarrow$ number of clusters
$\mathbf{X} \leftarrow$ data
$c_1, \ldots, c_k \leftarrow \mathbf{X}$ ▷ Initialize random clusters
repeat
  $c'_1, \ldots, c'_k \leftarrow c_1, \ldots, c_k$
  for $i \in \{1, \ldots, k\}$ **do**
    $centroid_i \leftarrow \frac{1}{n} \sum_{j=1}^{n} x_j, \quad x \in c_i$ ▷ Calculate the centroids
  end for
  $min\_dist = \infty$ ▷ Initialize minimum distance to a large number
  for $x \in \mathbf{X}$ **do**
    for $i \in \{1, \ldots, k\}$ **do**
      if $||x - centroid_i||_2^2 \leq min\_dist$ **then**
        $c_i \leftarrow x$ ▷ Assign x to the closest cluster
        $min\_dist \leftarrow ||x - centroid_i||_2^2$
      end if
    end for
  end for
until $c_1, \ldots, c_k = c'_1, \ldots, c'_k$ ▷ End when the clusters do not change

Variations to this algorithm are possible by, for example, using another distance metric such as Manhattan distance, or by using actual datapoints as representatives instead of virtual centroids (k-medoids). The k-means++ -algorithm (Arthur and Vassilvitskii 2007) used in this research has the optimization steps as the Lloyd's algorithm, but initializes the starting

state by sampling the centroids randomly from datapoints with probability proportional to the distance squared from the closest already sampled centroid, with the first centroid chosen randomly.

The minimization of intra-cluster distances leads to the division of the datapoints into Voronoi cells. This means the k-means clustering assumes the clusters to be roughly spherical and equidistant. The number of clusters is also fixed. For these reasons k-means clustering alone is not enough in explorative data-analysis.

5.2.2 OPTICS

OPTICS (Ankerst et al. 1999) as a density-based algorithm which attempts to find clusters where some minimum amount of datapoints are inside a region of a certain size, i.e. clusters with some density threshold. This allows the clusters to differ in shapes and sizes, and the number of the clusters needs not to be defined in advance.

Clustering using the OPTICS algorithm consists of two key parts, calculating reachability values for each of the datapoints (this is the OPTICS algorithm in the original paper), and using the reachability values to construct the clustering structure. The reachability values are a ordered and when graphed the dense areas in the data are represented as valleys in the visualization, with cluster boundaries being the cluster boundaries or noise. The OPTICS algorithm is described in 2

If a fixed threshold ε below which a point is considered a member of a cluster, results similar to another clustering algorithm, DBSCAN, are reached (Ankerst et al. 1999). However, in the original paper a more automatic technique of finding ξ-clusters is introduced: A ξ-cluster is an area $[a,b]$ in the ordered set of reachability values which is larger than the minimum amount of datapoints given as a hyperparameter, with the edges defined by the hyperparameter ξ which determines how large the change in reachability values needs to be for a point to be considered either a or b.

Algorithm 2 OPTICS

$minPts \leftarrow$ minimum number of neighboring points
$\varepsilon \leftarrow$ "generating distance"
$\{x_1, \ldots, x_n\} \leftarrow$ data
$reachability\ values \leftarrow \{\}$ ▷ Initialize output as an empty set
$priority - queue \leftarrow \{\}$
add random datapoint p to priority-queue as (p, ∞)
while priority-queue not empty **do**
    $(p, reachability) \leftarrow$ first element of the priority-queue
    $\varepsilon' \leftarrow$ minimum distance s.t. the neighborhood of p contains $minPts$-points
    for each x with distance$(p,x) \leq \varepsilon$ **do**
        $reachability_p \leftarrow max(\text{distance}(p, \varepsilon'), \text{distance}(p,x))$
        if $x \in reachability\ values$ **then**
            break
        else if $x \in$ priority-queue **then**
            Update $(x, reachability_x) \leftarrow (x, min(reachability_x, reachability_p))$
        else
            add $(x, reachability_p)$ to priority-queue
        end if
    end for
    add $(p, reachability)$ to $reachability\ values$
    Substitute first element of priority-queue with max reachability element
    Re-order priority queue
end while
Return $reachability\ values$

5.3 Cluster evaluation

The final stage before the interpretation of results in cluster analysis is the evaluation of results of the clustering (Halkidi, Batistakis, and Vazirgiannis 2001). The methods used in this research can be divided into internal and external evaluation.

In internal evaluation the clustering results are evaluated using the features of the clustered

dataset itself. In this research only such method used was the calculation of silhouette scores to compare the performances of k-means clustering when increasing the number of dimensions. Silhouette scores are calculated for each datapoint i by taking the mean distance $a(i)$ to other points in its cluster and the mean distance to the points in the closest different cluster $b(i)$. Silhouette score can then be calculated as

$$s(i) = \frac{b(i) - a(i)}{max\{a(i), b(i)\}}.$$

This results in a silhouette score between $[0, 1]$, where larger values are desired and negative values indicate errors in cluster assignments.

In external evaluation some additional info outside of the clustered data is used. This can include known labels to test the validity of clustering. For example, in this research the labels and datasets for the clustered image data were known, but only the raw pixel data were used for the clustering. The method used to evaluate the success in this research was the adjusted Rand index (Hubert and Arabie 1985). The original Rand index is calculated as

$$RI = \frac{a+d}{a+b+c+d}$$

where (in the case of a classification problem) a is the number of true positives, b is the number of false positives, c is the number of false negatives, and d is the number of true negatives, and the value is in the interval $[0, 1]$ The adjusted Rand index was developed to counter the problems arising from randomness in the original, as random classification does not produce a constant Rand index (Santos and Embrechts 2009). The adjusted Rand index has also a maximum value of 1, but the expected value of a random classification is 0. Adjusted Rand index can be calculated as

$$ARI = \frac{\binom{n}{2}(a+d) - [(a+b)(a+c)+(c+d)(b+d)]}{\binom{n}{2}^2 - [(a+b)(a+c)+(c+d)(b+d)}.$$

6 Data and methods

In this chapter the datasets used are described in detail, including the preprocessing of the data. The tools and specific methods used are also presented, with an emphasis on practical implementation in contrast to the previous chapters.

6.1 Datasets used

The FakeKnee -models were trained on radiograph images from the Osteoarthritis Iniative (OAI) dataset. Synthetic images generated by the model were then compared with radiograph images from Multicenter Osteoarthritis Study (MOST). These three datasets are detailed in the following subsections.

6.1.1 OAI and Synthetic images

The Osteoarthritis Initiative (OAI) was an observational study of knee osteoarthritis (Nevitt, Felson, and Lester 2006). The data were collected from 4796 participants aged 45-79 and included X-ray and MRI images, and biosamples. The X-ray images used in this research and the training of the FakeKnee -models 4.1 were obtained from a previous study (Chen et al. 2019) to automatically identify the knee joint area from leg radiographs in the OAI dataset. The images were divided by Kellgren-Lawrence graded into two groups KL01 and KL234 with grades 0-1 and 2-4 respectively.

To standardize the orientation the images of the left knee were mirrored horizontally and negative channel images were inverted. All of the images were contrast-equalized using histogram equalization, and scaled down from $299 \times 299 \times 1$ to $210 \times 210 \times 1$. Blurry images were then removed by filtering the image using a Laplacian kernel and calculating the variance of the resultant image and discarding images with variance < 350. Finally, 38 images with artifacts such as surgical prosthestics or scratches in the X-ray were removed manually. When training the FakeKnee-models the class KL01 has 3205 images and the class KL234 had 2351 images. This was further reduced two 2000 images in each class by random sampling to have balanced datasets in this research. The trained FakeKnee-models were the used

to generate 2000 images in each class. No further preprocessing was done to these images.

6.1.2 MOST

Multicenter Osteoarthritis Study was a longitudinal study conducted in the United States of America beginning in 2003 with baseline assessments and continuing for seven years with regular follow-up assessments. The assessments consisted of (but not limited to) joint space narrowing, formation of osteophytes, and Kellgren-Lawrence grading from radiological data (X-ray and MRI) of lateral, posterior-anterior, and full limb images. The baseline assessments had 3026 observations with the amount decreasing slightly in the follow-up assessments.

For the purposes of this thesis only the baseline KL-grading and posterior-anterior X-ray images were considered. This was done to maximize the amount of participants without including a participant's X-ray and assessments twice. This limited the dataset to 3016 participants' X-ray images with individual KL-grading of both of their knees, a total of 6032 images.

The individual knees were extracted using BoneFinder, a tool for finding bone contours using random forest regression voting (Lindner et al. 2013). These contours were used to crop the extracted knees to match the synthetic and OAI images. If BoneFinder failed to recognize a knee, the image was discarded. This happened on low quality images, or when the participant had had a knee surgery. This did not have an adverse effect as these images would have been removed at a later point.

Cropped images of the left knee were mirrored horizontally to match the FakeKnee models, which generates images of right knees. Histogram equalization was done to the images to equalize their contrast to match the OAI and generated images. The images were also categorized by the presence of osteoarthritis using the KL-grades; knees graded 0-1 being healthy and 2-4 being knees with osteoarthritis. These categories had 3531 and 2331 images respectively after this.

Radiographs with anomalies were then manually removed from the dataset. This included images with artifacts such as metal implants or from the scanning process of the original.

Examples of such images can be seen in figure 6. It has to be noted that most of the failed images were discarded when selecting the original 3016 subjects, as only knees with a valid KL-grading were selected. This manual process removed 87 and 57 from the respective classes.

The remaining images were evaluated for their blurriness by filtering the image using a Laplacian kernel and calculating the variance of the resultant image. The images with the highest variance were used to create the final dataset with 2000 images healthy knees and 2000 with various stages of osteoarthritis to ensure balanced categories.

Figure 6. A normal image, an image with an artifact, and an image with text overlap

6.2 Experiment details

In order to analyze the images and how they cluster the following experiment (figure 7) was conducted. First, the features were extracted from both real (MOST) and synthetic (FakeKnee) imagesets using a pretrained VGG19. These features which are 512-dimensional vectors were then projected to a lower dimension by using principal component analysis (PCA, 5.1.1) or uniform manifold approximation and projection (UMAP, 5.1.2). The projection to a lower dimension using PCA is a linear operation, and can be inserted into a fully connected neural network layer as such, which was used later in Grad-CAM analysis. Multiple different levels of dimension reduction were tested to find if different cluster structures emerge when varying the complexity of the features.

The features, which were embedded in fairly few (≤ 30) dimensions, were clustered using either k-means clustering or OPTICS depending on the distribution of the embedded features. The parameters of these algorithms (number of clusters/centroids in k-means, minimum clus-

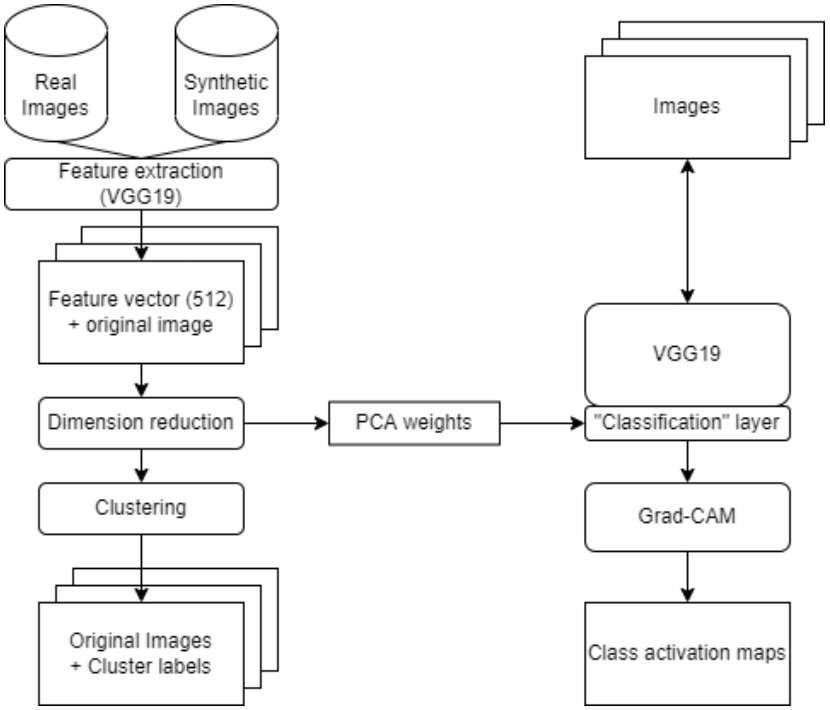

Figure 7. The experiment pipeline

ter members and ξ-value in OPTICS) were varied to find meaningful cluster structures. Detailed descriptions of k-means and OPTICS can be found in chapters 5.2.1 and 5.2.2. The clusters were analyzed to answer the following questions:

- Is there a difference in the features between synthetic and real images?
- Are clearly failed synthetic images distinguishable from convincing ones using this method?

A selection of embedded feature vectors with their corresponding images was chosen for the Grad-CAM analysis. The vectors were selected based on the clustering results, and were in practice either outliers or cluster representatives (i.e. centroids). The corresponding images of these feature vectors were then fed to VGG19 with the same weights as before, but with the coefficients obtained from principal component analysis as the final layer, which is usually used for classification. This was done to acquire the gradients between the embedded feature vector and the layers of VGG19, highlighting the parts of the image which contributed most to the placement of the vector in the clustering.

The experiment was replicated using OAI-images to verify the results. This part was identical to the main one, with the exception of having OAI-images in the place of synthetic images. The results were then compared to show the similarities between the clusterings, and the differences between the datasets.

7 Results

The synthetic images were found to be easily distinguishable from images from the MOST-dataset using cluster analysis. However the OAI images used to train the FakeKnee model clustered very similarly. Grad-CAM revealed mainly that in sharper images the "attention" of the convolutional neural network is focused on smaller details.

7.1 Cluster analysis

Reducing the dimension to two using PCA and plotting the resulting embedded feature vectors produces figure 8. In it the MOST -images can be seen to form roughly two dense clusters and a sparse cluster. The synthetic images from the FakeKnee-model forms a one large cluster which mixes a with one of the MOST-clusters. The cumulative explained variance ratio of this projection was 0.495.

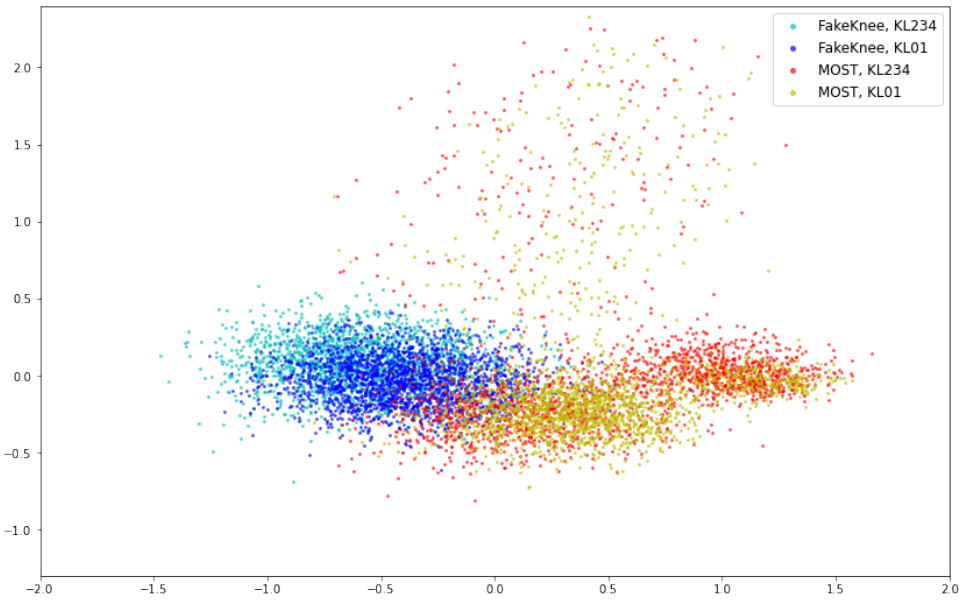

Figure 8. MOST and synthetic images projected to 2D using PCA

The suggested number of clusters for k-means (using inertia values and the elbowing method) was four. This clustering can be seen in figure 9. The 4-cluster k-means separated the synthetic from MOST-images with some success, cluster 4 being 83.0% synthetic images,

and clusters 1 and 2 being 96.6% and 100.0% real images respectively.

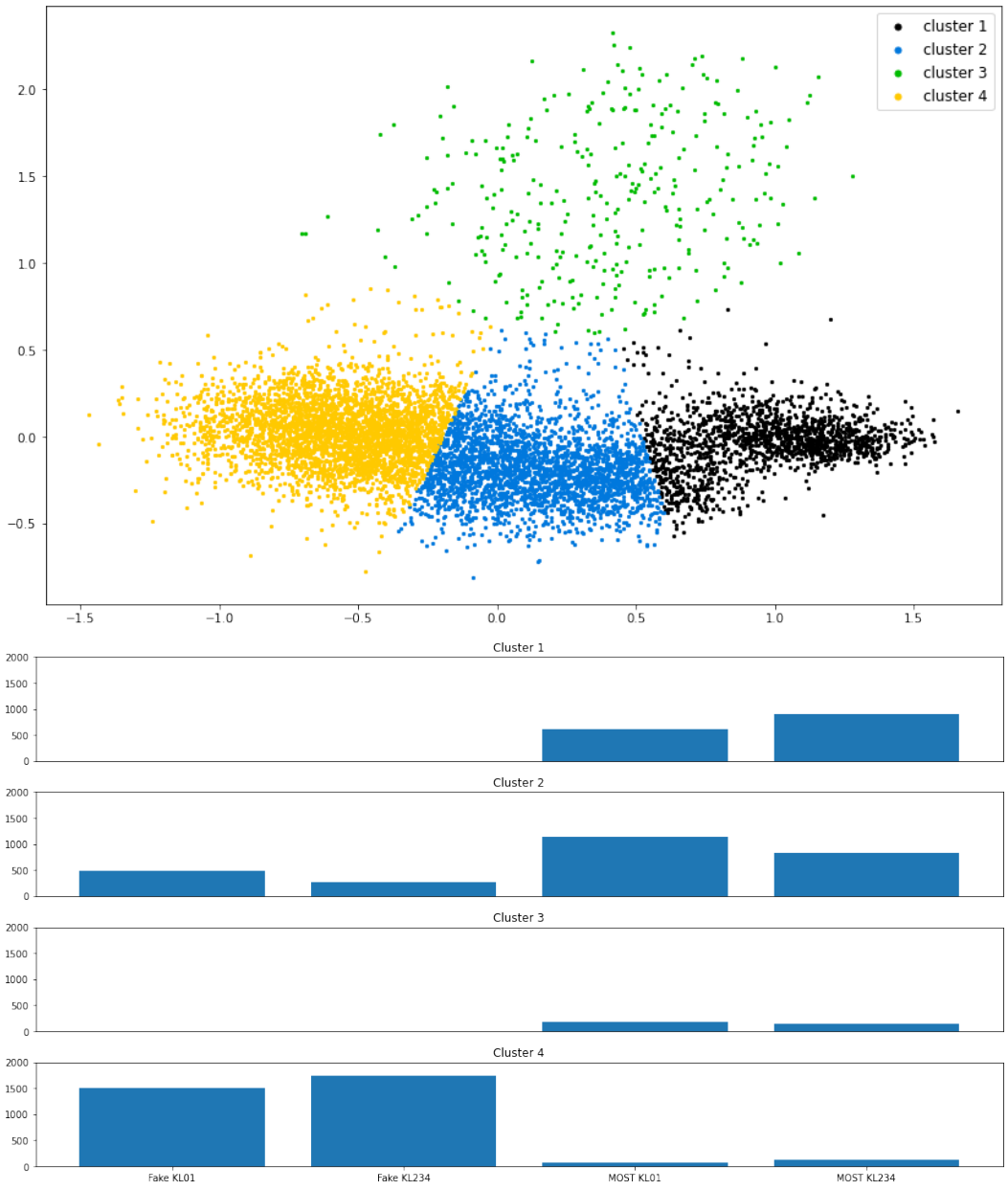

Figure 9. K-means clustering of MOST and synthetic with four clusters in 2D

A plot of the explained variances can be seen in figure 10. Cumulative explained variance ratio of 0.90 was achieved with 30 principal components. Principal component analysis with more components and using k-means clustering changes the cluster structure only a little after 4 components: The adjusted Rand indices between embeddings with more components are close to 1. The adjusted Rand indices are shown in table 1. The optimal number of

clusters seemed to approach 4, but the delta between concurrent inertia values gets generally smaller when increasing the components used. The silhouette value averages also tended to lower values which could indicate worse clusterings when the number of components gets larger. This knowledge guided the selection of the latent dimensions used in further analysis, these dimensions being 2, 3, 4, 5, 10, 20 and 30.

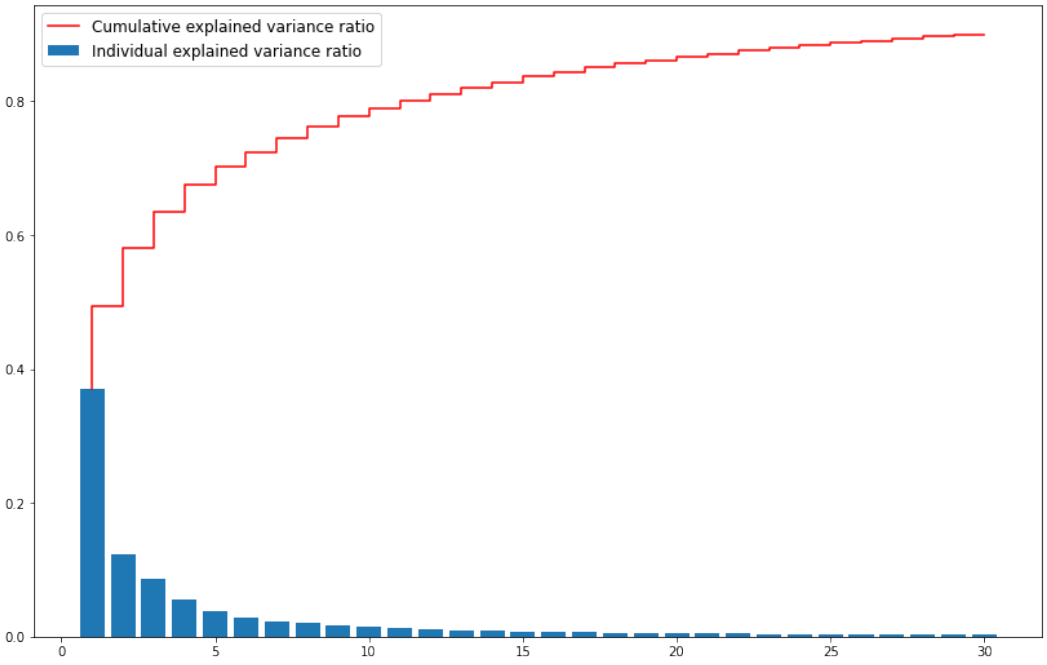

Figure 10. Explained variance ratios of principal component analysis

7.1.1 UMAP embeddings clustering using OPTICS

The UMAP assisted clustering using OPTICS performed better than the PCA and k-means -method on separating the synthetic from the real images. This can be seen in figure 11 which shows that a 2D projection using UMAP forms distinct partitions in the data. This, in contrast to the 2D projection using PCA (figure 8), can be more easily clustered into synthetic and real images.

Clustering these 2D embeddings using OPTICS ($minPts = 50, \xi = 0.1$) produced four clusters, with a region between clusters 2 and 3 labeled as noise (figure 12). The distributions of original labels in these clusters are in table 2. The synthetic images (with some exceptions) were clustered into one cluster, and the images from MOST were clustered into three clusters

Table 1. Adjusted Rand indices between k-means clusterings ($k = 4$) of the selected PCA components (dimensions)

components	2	3	4	5	10	20	30
2	1.0	0.750	0.723	0.720	0.712	0.714	0.713
3		1.0	0.923	0.919	0.909	0.912	0.909
4			1.0	0.988	0.969	0.972	0.970
5				1.0	0.977	0.977	0.976
10					1.0	0.994	0.995
20						1.0	0.997
30							1.0

Table 2. Distributions of images in clusters (UMAP 2D/OPTICS)

	MOST		Synthetic	
cluster	KL01	KL234	KL01	KL234
1	0	2	1995	1996
2	1367	940	4	1
3	172	103	0	0
4	344	849	1	3

of varying sizes. There was seemingly no clear clustering by KL-grades.

The clustering method had little effect on the labels of the 2D features. The adjusted Rand index between the k-means and OPTICS labels was 0.995 on on $k = 4$ and OPTICS with $minPts = 50$ and $\xi = 0.1$. However on higher dimensionalities OPTICS with the same hyperparameters finds multiple smaller clusters: Twenty-three subclusters were discovered in UMAP embeddings with 30 features. These are shown in figure 13 as a 2D projection, and were used as the basis for Grad-CAM visualizations in chapter 7.2.

7.1.2 Validation using OAI

The success in generating synthetic versions of OAI images was evaluated by replicating the previous clusterings using OAI images in place of the synthetic FakeKnee images, and

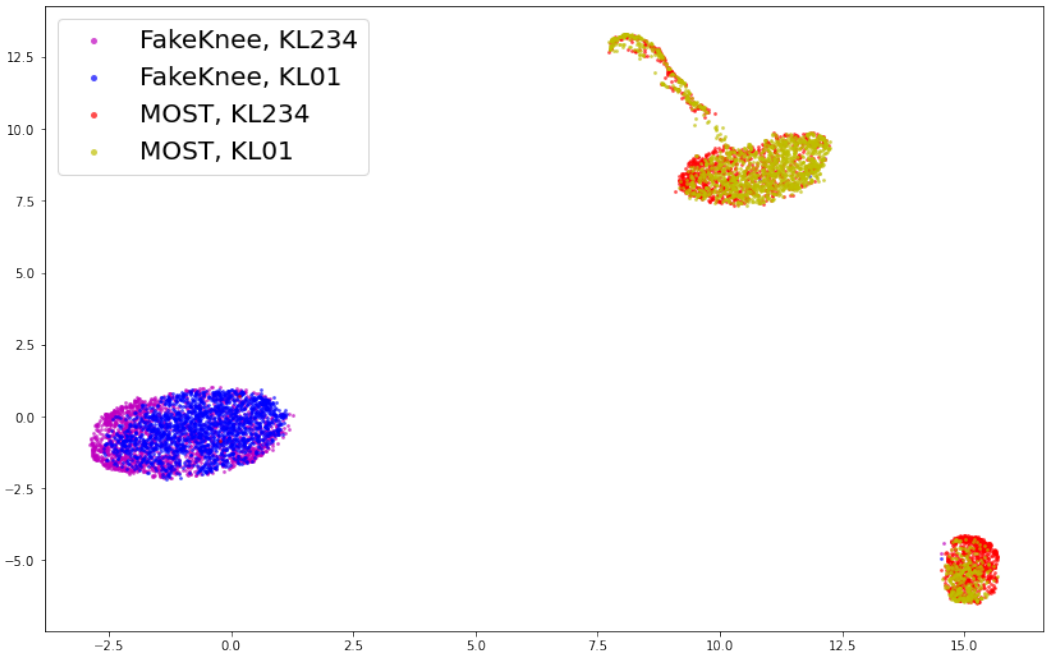

Figure 11. MOST and synthetic images projected to 2D using UMAP

comparing the clusterings. The same MOST images were used, and the new clusterings were done independent of the original clusterings. The PCA projection of MOST and OAI can be seen in figure 14.

While there is a lot of similarity between the distributions of the OAI and synthetics images as seen in figures 8 and 14, clustering them into four clusters using k-means gives an adjusted Rand index of 0.465. However, as there is no clear cluster structures visible and the assumptions of k-means clustering are not fulfilled, the features are divided into four similarly sized segments and there could be a lot of difference in labels near the edges of the segments.

Embedding the MOST and OAI images using UMAP to two dimensions forms a similar structure to figure 11, and these features can be successfully clustered using OPTICS (minPts = 50, $\xi = 0.1$), as was done with MOST and synthetic images. The resulting clustering can be seen in figure 15, and the distribution of features are in table 3.

Using the embeddings (in dimensions 2, 3, 4, 5, 10, 20, and 30) produced similar clusters to ones obtained when comparing synthetic and MOST images. The adjusted Rand indices between the original clusters and validations clusters were over 0.95 on all embedding di-

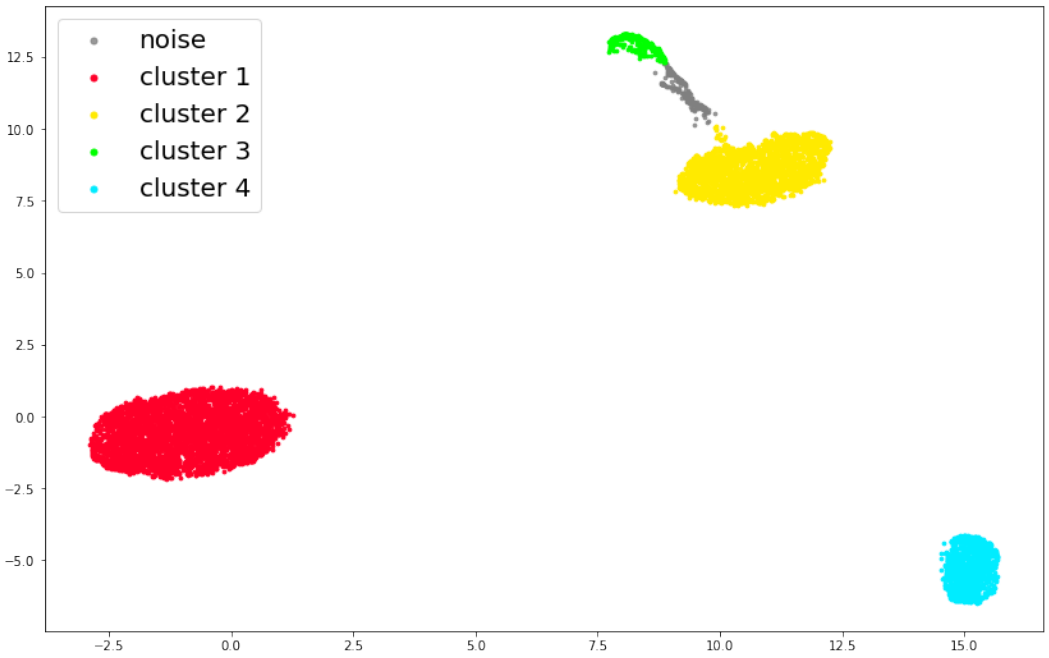

Figure 12. MOST and synthetic UMAP 2D embeddings clustered using OPTICS

Table 3. Clustering results of MOST and OAI images in 2D

	MOST		OAI	
cluster	KL01	KL234	KL01	KL234
0	15	13	1944	19805
1	1354	935	50	6
2	287	206	0	1
3	343	846	5	13

mensions from 2 onward, with 4D being the highest of these with ARI of 0.952 With few exceptions the OAI images and synthetic images were indistinguishable from each other using this method.

7.2 Grad-CAM visualizations

The subcluster representatives from clusters shown in figure 13 were calculated as the datapoint closest (w.r.t. their l^2 distance) to the mean of all datapoints in a cluster. While UMAP is not a linear transformation and thus the datapoint closest to the mean is not necessarily in

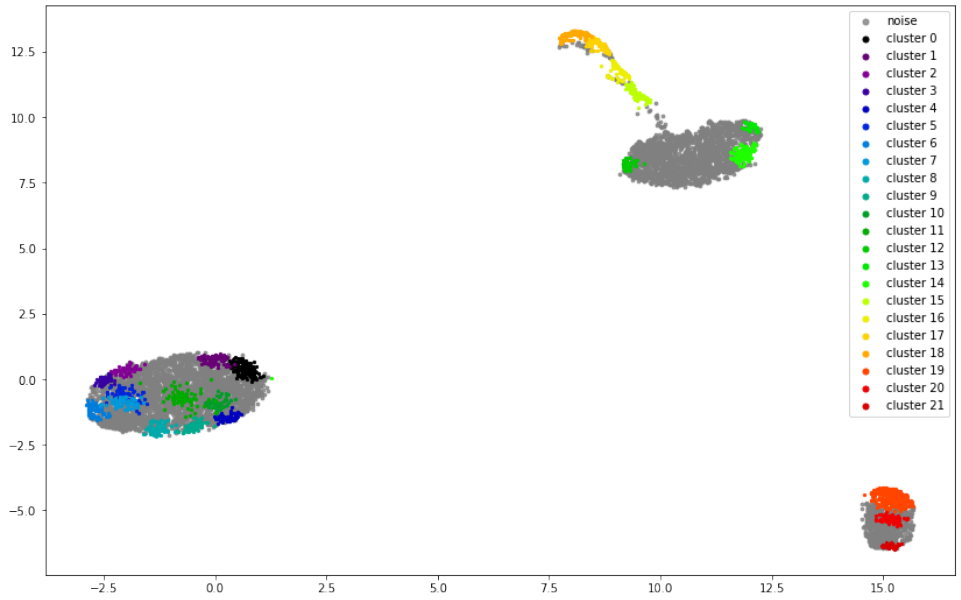

Figure 13. Subclusters discovered in 30d

the middle of a cluster, the cluster sizes were small and their distances relative to one another were big enough for this to have little effect.

The Grad-CAM results show no clear differences between the clusters. The clusters 0-11 were mostly synthetic images and the Grad-CAM visualizations show that the VGG19 - network focused on sharp edges, and put emphasis on the joint space an tibial spike, while also focusing on the femur and patella. The clusters 12-14 were real images and while clearly distinct from the synthetic images in the UMAP projection, they were the closest when using PCA. The Grad-CAM heatmaps could also indicate this, although the difference is not clear.

Subclusters 15-18 were part of the smallest of the clusters. A noticeable "striping" could be seen in all of the images in the clusters. The heatmaps show that the network was not totally distracted by this features, but as they were distinctly clustered together, this might have been the most prominent feature.

Subclusters 19-21 were a part of the final larger cluster. This cluster consisted of the blurrier image from MOST. The heatmaps seem to light smaller areas, and this could possibly be due to the network struggling to find distinct features.

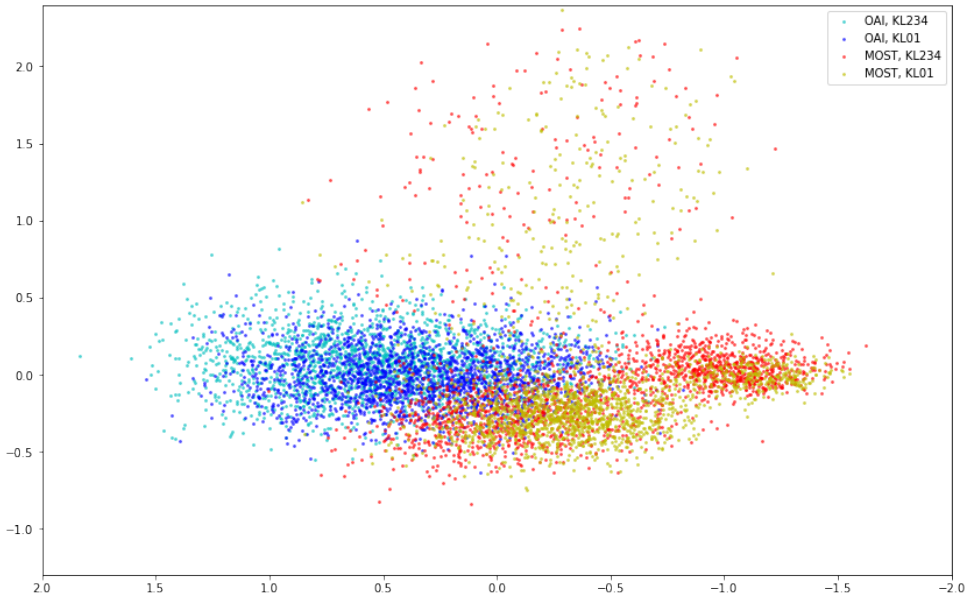

Figure 14. MOST and OAI images projected to 2D using PCA

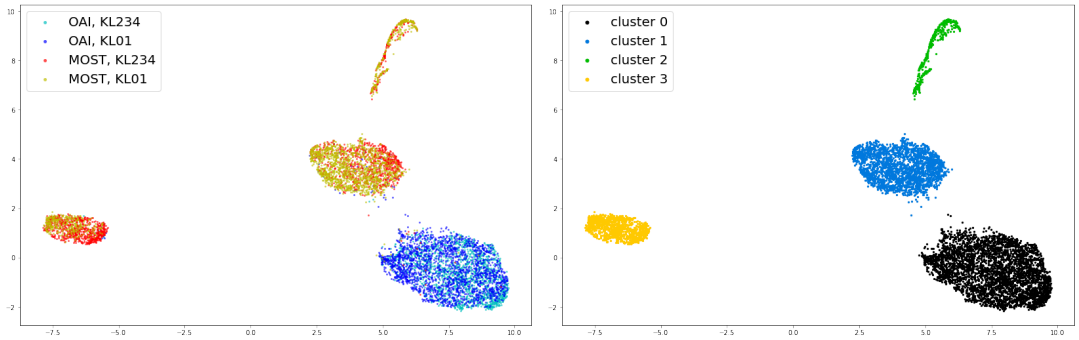

Figure 15. MOST and OAI images projected to 2D using UMAP and clustering using OPTICS

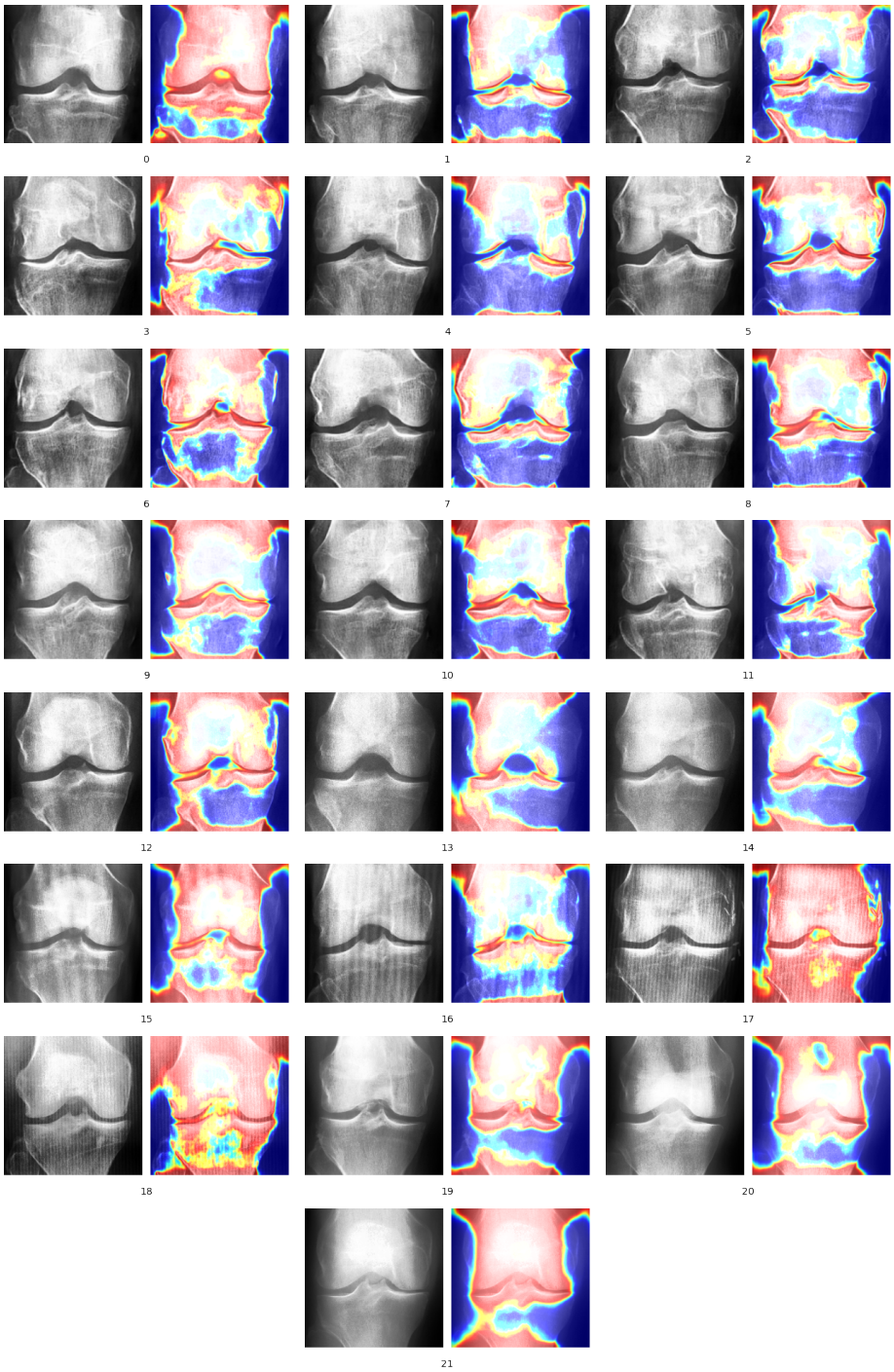

Figure 16. Grad-CAM visualizations of the 22 cluster representatives. Two copies of the same X-ray are shown for each sample, with the original and one with Grad-CAM heatmap superimposed on it. The red sections of the image contributed more to the latent embedding of the image.

8 Conclusion

In this Master's thesis synthetic images created by Prezja et al. 2022 were studied by feature extraction and cluster analysis methods and comparing the results to real X-ray images.

Using UMAP together with OPTICS successfully separated the synthetic images from the real images of MOST dataset. However, using the same method with the original OAI training images (instead of synthetic) yielded similar results as the OAI and MOST images separated as well. This indicates the success of the synthetic images capturing the distribution of the training set.

The method used to extract features showed to be susceptible to "unimportant" features (stripes, blurriness), as they dominated the clustering results. This might be alleviated by fine-tuning the models to ignore these features. Denoising autoencoders could also be used to find better embeddings while making them more robust.

The variations between datasets was clearly shown. While the preprocessing done to both datasets was as similar as possible, the differences between them were shown in the clustering which almost perfectly separated them. This result shows the need for diverse training sets for computer vision applications. Models tested on images from the same distribution as they were trained on will give a too optimistic measure of their capabilities. For example, a model trained on OAI images (or synthetic images) would not perform as well on MOST images, and therefore would probably struggle in hospital scenarios.

There were some indications of KL-grades affecting the embedding. Most of the knees graded 2, 3 and 4 tended to be on the opposite side of the cluster with regards to the knees graded 0 or 1. The huge amount of overlap between these two classes does point to the difficulty of the task of automatically grading the severity of osteoarthritis.

The results from Grad-CAM analysis were unclear. While the method indicated that the feature extraction focused on the essential locations on the images, the methods for analyzing the result would need to be more refined. Further development is needed for making these unsupervised methods more explainable.

Bibliography

Abadi, Martín, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, et al. 2015. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems.* www.tensorflow.org..

Ankerst, Mihael, Markus M Breunig, Hans-Peter Kriegel, and Jörg Sander. 1999. "OPTICS: Ordering points to identify the clustering structure". *ACM Sigmod record* 28 (2): 49–60.

Arjovsky, Martin, and Léon Bottou. 2017. "Towards principled methods for training generative adversarial networks". *arXiv preprint arXiv:1701.04862.*

Arjovsky, Martin, Soumith Chintala, and Léon Bottou. 2017. "Wasserstein generative adversarial networks". In *International conference on machine learning,* 214–223. PMLR.

Arthur, David, and Sergei Vassilvitskii. 2007. "K-means++ the advantages of careful seeding". In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms,* 1027–1035.

Becht, Etienne, Leland McInnes, John Healy, Charles-Antoine Dutertre, Immanuel WH Kwok, Lai Guan Ng, Florent Ginhoux, and Evan W Newell. 2019. "Dimensionality reduction for visualizing single-cell data using UMAP". *Nature biotechnology* 37 (1): 38–44.

Bishop, Christopher M, and Nasser M Nasrabadi. 2006. *Pattern recognition and machine learning.* Volume 4. 4. Springer.

Chen, Pingjun, Linlin Gao, Xiaoshuang Shi, Kyle Allen, and Lin Yang. 2019. "Fully automatic knee osteoarthritis severity grading using deep neural networks with a novel ordinal loss". *Computerized Medical Imaging and Graphics* 75:84–92.

Choi, Edward, Siddharth Biswal, Bradley Malin, Jon Duke, Walter F Stewart, and Jimeng Sun. 2017. "Generating multi-label discrete patient records using generative adversarial networks". In *Machine learning for healthcare conference,* 286–305. PMLR.

Clevert, Djork-Arné, Thomas Unterthiner, and Sepp Hochreiter. 2015. "Fast and accurate deep network learning by exponential linear units (ELUs)". *arXiv preprint arXiv:1511.07289.*

Deng, Jia, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ”Imagenet: A large-scale hierarchical image database”. In *2009 IEEE conference on computer vision and pattern recognition,* 248–255. Ieee.

Diaz-Papkovich, Alex, Luke Anderson-Trocmé, Chief Ben-Eghan, and Simon Gravel. 2019. ”UMAP reveals cryptic population structure and phenotype heterogeneity in large genomic cohorts”. *PLoS genetics* 15 (11): e1008432.

Dong, Wei, Charikar Moses, and Kai Li. 2011. ”Efficient k-nearest neighbor graph construction for generic similarity measures”. In *Proceedings of the 20th international conference on World wide web,* 577–586.

Drineas, Petros, Alan Frieze, Ravi Kannan, Santosh Vempala, and Vishwanathan Vinay. 2004. ”Clustering large graphs via the singular value decomposition”. *Machine learning* 56:9–33.

European Parliament, Council of the European Union. 2016. *Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation).* Official Journal of the European Union. https://eur-lex.europa.eu/eli/reg/2016/679/oj.

Fukushima, Kunihiko. 1980. *Biological Cybernetics Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position.*

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning.* Http://www.deeplearningbook.org. MIT Press.

Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. ”Generative Adversarial Nets”. In *Advances in Neural Information Processing Systems,* edited by Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, volume 27. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf.

Gulrajani, Ishaan, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. 2017. "Improved training of Wasserstein GANs". *Advances in neural information processing systems* 30.

Halkidi, Maria, Yannis Batistakis, and Michalis Vazirgiannis. 2001. "On clustering validation techniques". *Journal of intelligent information systems* 17:107–145.

He, Kaiming, Ross Girshick, and Piotr Dollár. 2019. "Rethinking imagenet pre-training". In *Proceedings of the IEEE/CVF International Conference on Computer Vision,* 4918–4927.

Health, U.S. Department of, and Human Services. 2000. *HIPAA privacy rule.* 45 CFR Parts 160 and 164. Standards for privacy of individually identifiable health information; final rule.

Hubert, Lawrence, and Phipps Arabie. 1985. "Comparing partitions". *Journal of classification* 2:193–218.

Hunter, David J, Deborah Schofield, and Emily Callander. 2014. "The individual and socioeconomic impact of osteoarthritis". *Nature Reviews Rheumatology* 10 (7): 437–441.

Jain, Anil K, and Richard C Dubes. 1988. *Algorithms for clustering data.* Prentice-Hall, Inc.

Joliffe, Ian T. 2002. *Principal Component Analysis.* Springer New York, NY. https://doi.org/10.1007/b98835.

Jung, Hyungsik, and Youngrock Oh. 2021. "Towards better explanations of class activation mapping". In *Proceedings of the IEEE/CVF International Conference on Computer Vision,* 1336–1344.

Kaila-Kangas, Leena. 2007. "Musculoskeletal disorders and diseases in Finland: results of the health 2000 survey".

Karbhari, Yash, Arpan Basu, Zong Woo Geem, Gi-Tae Han, and Ram Sarkar. May 2021. "Generation of Synthetic Chest X-ray Images and Detection of COVID-19: A Deep Learning Based Approach". *Diagnostics* 11, number 5 (): 895. ISSN: 2075-4418, visited on April 19, 2023. https://doi.org/10.3390/diagnostics11050895. https://www.mdpi.com/2075-4418/11/5/895.

Käypä hoito. 2016. ”Knee and hip osteoarthritis. Current Care Guideline”. Visited on June 2, 2023. www.kaypahoito.fi.

Kellgren, Jonas H, and JS1006995 Lawrence. 1957. ”Radiological assessment of osteo-arthrosis”. *Annals of the rheumatic diseases* 16 (4): 494.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton. 2017. ”Imagenet classification with deep convolutional neural networks”. *Communications of the ACM* 60 (6): 84–90.

Landau, Sabine, Morven Leese, Daniel Stahl, and Brian S Everitt. 2011. *Cluster analysis.* John Wiley & Sons.

LeCun, Yann, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. ”Gradient-based learning applied to document recognition”. *Proceedings of the IEEE* 86 (11): 2278–2324.

Lee, Lok Sze, Ping Keung Chan, Chunyi Wen, Wing Chiu Fung, Amy Cheung, Vincent Wai Kwan Chan, Man Hong Cheung, Henry Fu, Chun Hoi Yan, and Kwong Yuen Chiu. 2022. ”Artificial intelligence in diagnosis of knee osteoarthritis and prediction of arthroplasty outcomes: a review”. *Arthroplasty* 4 (1): 16.

Lespasio, Michelle J, Nicolas S Piuzzi, M Elaine Husni, George F Muschler, AJ Guarino, and Michael A Mont. 2017. ”Knee osteoarthritis: a primer”. *The Permanente Journal* 21.

Lindner, C., S. Thiagarajah, J. M. Wilkinson, The arcOGEN Consortium, G. A. Wallis, and T. F. Cootes. 2013. ”Fully Automatic Segmentation of the Proximal Femur Using Random Forest Regression Voting”. *IEEE Transactions on Medical Imaging* 32 (8): 1462–1472. https://doi.org/10.1109/TMI.2013.2258030.

Litjens, Geert, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen A.W.M. van der Laak, Bram van Ginneken, and Clara I. Sánchez. December 2017. ”A survey on deep learning in medical image analysis”. *Medical Image Analysis* 42 (): 60–88. ISSN: 13618415, visited on April 12, 2023. https://doi.org/10.1016/j.media.2017.07.005.

Lloyd, Stuart. 1982. ”Least squares quantization in PCM”. *IEEE transactions on information theory* 28 (2): 129–137.

McInnes, Leland, John Healy, and James Melville. 2020. *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction.* arXiv: 1802.03426 [stat.ML].

Murtaza, Hajra, Musharif Ahmed, Naurin Farooq Khan, Ghulam Murtaza, Saad Zafar, and Ambreen Bano. 2023. "Synthetic data generation: State of the art in health care domain". *Computer Science Review* 48:100546.

Narayanan, Arvind, and Edward W Felten. 2014. "No silver bullet: De-identification still doesn't work". *White Paper* 8.

Nevitt, M, D Felson, and Gayle Lester. 2006. "The osteoarthritis initiative". *Protocol for the cohort study* 1.

Phillips, P Jonathon, Carina A Hahn, Peter C Fontana, David A Broniatowski, and Mark A Przybocki. 2020. "Four principles of explainable artificial intelligence". *Gaithersburg, Maryland,* 18.

Prezja, Fabi, Juha Paloneva, Ilkka Pölönen, Esko Niinimäki, and Sami Äyrämö. 2022. "DeepFake knee osteoarthritis X-rays from generative adversarial neural networks deceive medical experts and offer augmentation potential to automatic classification". *Scientific Reports* 12 (1): 18573. ISSN: 2045-2322, visited on April 12, 2023. https://doi.org/10.1038/s41598-022-23081-4. https://www.nature.com/articles/s41598-022-23081-4.

Roos, Ewa M, and Nigel K Arden. 2016. "Strategies for the prevention of knee osteoarthritis". *Nature Reviews Rheumatology* 12 (2): 92–101.

Rosenblatt, Frank. 1958. "The perceptron: a probabilistic model for information storage and organization in the brain." *Psychological review* 65 (6): 386.

Samala, Ravi K, Heang-Ping Chan, Lubomir Hadjiiski, Mark A Helvie, Jun Wei, and Kenny Cha. 2016. "Mass detection in digital breast tomosynthesis: Deep convolutional neural network with transfer learning from mammography". *Medical physics* 43 (12): 6654–6666.

Santos, Jorge M, and Mark Embrechts. 2009. "On the use of the adjusted rand index as a metric for evaluating supervised classification". In *Artificial Neural Networks–ICANN 2009: 19th International Conference, Limassol, Cyprus, September 14-17, 2009, Proceedings, Part II 19,* 175–184. Springer.

Selvaraju, Ramprasaath R, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. ”Grad-cam: Visual explanations from deep networks via gradient-based localization”. In *Proceedings of the IEEE international conference on computer vision,* 618–626.

Shin, Hoo Chang, Holger R. Roth, Mingchen Gao, Le Lu, Ziyue Xu, Isabella Nogues, Jianhua Yao, Daniel Mollura, and Ronald M. Summers. May 2016. ”Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning”. *IEEE Transactions on Medical Imaging* 35 (5): 1285–1298. ISSN: 1558254X. https://doi.org/10.1109/TMI.2016.2528162.

Simonyan, Karen, and Andrew Zisserman. 2014. ”Very deep convolutional networks for large-scale image recognition”. *arXiv preprint arXiv:1409.1556.*

Sinusas, Keith. 2012. ”Osteoarthritis: diagnosis and treatment”. *American family physician* 85 (1): 49–56.

Tiulpin, Aleksei, Jérôme Thevenot, Esa Rahtu, Petri Lehenkari, and Simo Saarakkala. 2018. ”Automatic knee osteoarthritis diagnosis from plain radiographs: a deep learning-based approach”. *Scientific reports* 8 (1): 1–10.

Van der Maaten, Laurens, and Geoffrey Hinton. 2008. ”Visualizing data using t-SNE.” *Journal of machine learning research* 9 (11).

Villani, Cédric, et al. 2009. *Optimal transport: old and new.* Volume 338. Springer.

Viola, Paul, and Michael J Jones. 2004. ”Robust real-time face detection”. *International journal of computer vision* 57:137–154.

Wieland, Heike A, Martin Michaelis, Bernhard J Kirschbaum, and Karl A Rudolphi. 2005. ”Osteoarthritis—an untreatable disease?” *Nature reviews Drug discovery* 4 (4): 331–344.

Zhou, Bolei, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. 2016. ”Learning deep features for discriminative localization”. In *Proceedings of the IEEE conference on computer vision and pattern recognition,* 2921–2929.