FACULTY OF ENGINEERING
AND ARCHITECTURE

Improving Pose Estimation on Art Collections with Style Transfer

Tristan Verheecke
Student number: 20043518

Supervisors: Prof. dr. ir. Dieter De Witte, Prof. dr. Steven Verstockt
Counsellors: Kenzo Milleville, Ravi Khatri

Master's dissertation submitted in order to obtain the academic degree of
Master of Science in Information Engineering Technology

Academic year 2023-2024

GHENT
UNIVERSITY

Preface

I've been interested in Art my entire life. In fact, I've a degree in the Fine Arts from LUCA School of Arts. There, I was known for my technological ability and one of my professors at the time asked me why I didn't do anything with that in my artworks. That remark has since stuck with me and was part of my motivation to apply for readmission for my Master of Science. With all the advancements in AI, I started thinking more and more about doing work with that. Like Matisse and Turner, I'm not satisfied with the tools available, but want to create my own.

It was therefor to my delight that I was able to work on this thesis which has provided me the opportunity to acquire more insight in the subject. I would like to thank my supervisors Dieter De Witte and Steven Verstockt for this wonderful opportunity, and my counsellor Kenzo Milleville for his great guidance. As well as all the other people at IDLab for their feedback. I also want to thank Karine Lacaracina, Lies Van De Cappelle and the other people at RMFAB for providing help with the artistic sensibilities of the thesis.

Enjoy the read,

Tristan Verheecke
Ghent, June 2023

Conference Paper Title*

*Note: Sub-titles are not captured in Xplore and should not be used

1st Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

2nd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

3rd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

4th Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

5th Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

6th Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

***Abstract*—This document is a model and instructions for LaTeX. This and the IEEEtran.cls file define the components of your paper [title, text, heads, etc.]. *CRITICAL: Do Not Use Symbols, Special Characters, Footnotes, or Math in Paper Title or Abstract.**

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

This document is a model and instructions for LaTeX. Please observe the conference page limits.

II. Ease of Use

A. Maintaining the Integrity of the Specifications

The IEEEtran class file is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

III. Prepare Your Paper Before Styling

Before you begin to format your paper, first write and save the content as a separate text file. Complete all content and organizational editing before formatting. Please note sections III-A–III-E below for more information on proofreading, spelling and grammar.

Keep your text and graphic files separate until after the text has been formatted and styled. Do not number text heads—LaTeX will do that for you.

Identify applicable funding agency here. If none, delete this.

A. Abbreviations and Acronyms

Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract. Abbreviations such as IEEE, SI, MKS, CGS, ac, dc, and rms do not have to be defined. Do not use abbreviations in the title or heads unless they are unavoidable.

B. Units

- Use either SI (MKS) or CGS as primary units. (SI units are encouraged.) English units may be used as secondary units (in parentheses). An exception would be the use of English units as identifiers in trade, such as “3.5-inch disk drive”.
- Avoid combining SI and CGS units, such as current in amperes and magnetic field in oersteds. This often leads to confusion because equations do not balance dimensionally. If you must use mixed units, clearly state the units for each quantity that you use in an equation.
- Do not mix complete spellings and abbreviations of units: “Wb/m^2” or “webers per square meter”, not “webers/m^2”. Spell out units when they appear in text: “. . . a few henries”, not “. . . a few H”.
- Use a zero before decimal points: “0.25”, not “.25”. Use “cm^3”, not “cc”.)

C. Equations

Number equations consecutively. To make your equations more compact, you may use the solidus (/), the exp function, or appropriate exponents. Italicize Roman symbols for quantities and variables, but not Greek symbols. Use a long dash rather than a hyphen for a minus sign. Punctuate equations with commas or periods when they are part of a sentence, as in:

$$a + b = \gamma \tag{1}$$

Be sure that the symbols in your equation have been defined before or immediately following the equation. Use “(1)”, not

"Eq. (1)" or "equation (1)", except at the beginning of a sentence: "Equation (1) is . . ."

D. LaTeX-Specific Advice

Please use "soft" (e.g., `\eqref{Eq}`) cross references instead of "hard" references (e.g., `(1)`). That will make it possible to combine sections, add equations, or change the order of figures or citations without having to go through the file line by line.

Please don't use the `{eqnarray}` equation environment. Use `{align}` or `{IEEEeqnarray}` instead. The `{eqnarray}` environment leaves unsightly spaces around relation symbols.

Please note that the `{subequations}` environment in LaTeX will increment the main equation counter even when there are no equation numbers displayed. If you forget that, you might write an article in which the equation numbers skip from (17) to (20), causing the copy editors to wonder if you've discovered a new method of counting.

BibTeX does not work by magic. It doesn't get the bibliographic data from thin air but from .bib files. If you use BibTeX to produce a bibliography you must send the .bib files.

LaTeX can't read your mind. If you assign the same label to a subsubsection and a table, you might find that Table I has been cross referenced as Table IV-B3.

LaTeX does not have precognitive abilities. If you put a `\label` command before the command that updates the counter it's supposed to be using, the label will pick up the last counter to be cross referenced instead. In particular, a `\label` command should not go before the caption of a figure or a table.

Do not use `\nonumber` inside the `{array}` environment. It will not stop equation numbers inside `{array}` (there won't be any anyway) and it might stop a wanted equation number in the surrounding equation.

E. Some Common Mistakes

- The word "data" is plural, not singular.
- The subscript for the permeability of vacuum μ_0, and other common scientific constants, is zero with subscript formatting, not a lowercase letter "o".
- In American English, commas, semicolons, periods, question and exclamation marks are located within quotation marks only when a complete thought or name is cited, such as a title or full quotation. When quotation marks are used, instead of a bold or italic typeface, to highlight a word or phrase, punctuation should appear outside of the quotation marks. A parenthetical phrase or statement at the end of a sentence is punctuated outside of the closing parenthesis (like this). (A parenthetical sentence is punctuated within the parentheses.)
- A graph within a graph is an "inset", not an "insert". The word alternatively is preferred to the word "alternately" (unless you really mean something that alternates).
- Do not use the word "essentially" to mean "approximately" or "effectively".
- In your paper title, if the words "that uses" can accurately replace the word "using", capitalize the "u"; if not, keep using lower-cased.
- Be aware of the different meanings of the homophones "affect" and "effect", "complement" and "compliment", "discreet" and "discrete", "principal" and "principle".
- Do not confuse "imply" and "infer".
- The prefix "non" is not a word; it should be joined to the word it modifies, usually without a hyphen.
- There is no period after the "et" in the Latin abbreviation "et al.".
- The abbreviation "i.e." means "that is", and the abbreviation "e.g." means "for example".

An excellent style manual for science writers is [7].

F. Authors and Affiliations

The class file is designed for, but not limited to, six authors. A minimum of one author is required for all conference articles. Author names should be listed starting from left to right and then moving down to the next line. This is the author sequence that will be used in future citations and by indexing services. Names should not be listed in columns nor group by affiliation. Please keep your affiliations as succinct as possible (for example, do not differentiate among departments of the same organization).

G. Identify the Headings

Headings, or heads, are organizational devices that guide the reader through your paper. There are two types: component heads and text heads.

Component heads identify the different components of your paper and are not topically subordinate to each other. Examples include Acknowledgments and References and, for these, the correct style to use is "Heading 5". Use "figure caption" for your Figure captions, and "table head" for your table title. Run-in heads, such as "Abstract", will require you to apply a style (in this case, italic) in addition to the style provided by the drop down menu to differentiate the head from the text.

Text heads organize the topics on a relational, hierarchical basis. For example, the paper title is the primary text head because all subsequent material relates and elaborates on this one topic. If there are two or more sub-topics, the next level head (uppercase Roman numerals) should be used and, conversely, if there are not at least two sub-topics, then no subheads should be introduced.

H. Figures and Tables

a) Positioning Figures and Tables: Place figures and tables at the top and bottom of columns. Avoid placing them in the middle of columns. Large figures and tables may span across both columns. Figure captions should be below the figures; table heads should appear above the tables. Insert figures and tables after they are cited in the text. Use the abbreviation "Fig. 1", even at the beginning of a sentence.

TABLE I
TABLE TYPE STYLES

Table Head	**Table Column Head**		
	Table column subhead	***Subhead***	***Subhead***
copy	More table copy[a]		

[a]Sample of a Table footnote.

Fig. 1. Example of a figure caption.

Figure Labels: Use 8 point Times New Roman for Figure labels. Use words rather than symbols or abbreviations when writing Figure axis labels to avoid confusing the reader. As an example, write the quantity “Magnetization”, or “Magnetization, M”, not just “M”. If including units in the label, present them within parentheses. Do not label axes only with units. In the example, write “Magnetization (A/m)” or “Magnetization {A[m(1)]}”, not just “A/m”. Do not label axes with a ratio of quantities and units. For example, write “Temperature (K)”, not “Temperature/K”.

ACKNOWLEDGMENT

The preferred spelling of the word “acknowledgment” in America is without an “e” after the “g”. Avoid the stilted expression “one of us (R. B. G.) thanks ...”. Instead, try “R. B. G. thanks...”. Put sponsor acknowledgments in the unnumbered footnote on the first page.

REFERENCES

Please number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use “Ref. [3]” or “reference [3]” except at the beginning of a sentence: “Reference [3] was the first ...”

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the abstract or reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors’ names; do not use “et al.”. Papers that have not been published, even if they have been submitted for publication, should be cited as “unpublished” [4]. Papers that have been accepted for publication should be cited as “in press” [5]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

REFERENCES

[1] G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.
[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
[3] I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
[4] K. Elissa, “Title of paper if known,” unpublished.
[5] R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.
[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
[7] M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.

IEEE conference templates contain guidance text for composing and formatting conference papers. Please ensure that all template text is removed from your conference paper prior to submission to the conference. Failure to remove the template text from your paper may result in your paper not being published.

Contents

List of Figures

List of Tables

List of Acronyms

A

AdaIN	Adaptive Instance Normalization , 13, 19
AFHQ	Animal Face High Quality ix, 19, 31
AIC-HKD	AI Challenger Human Keypoint Detection , 4
AP	Average Precision , 10
AR	Average Recall , 10, 11
ASMs	Active Shape Models , 3
AST-IQAD	Arbitrary Style Transfer Image Quality Assessment Database , 27

B

BN	Batch Normalization viii, 11, 12

C

CBIR	Content Based Image Retrieveal viii, ix, 16, 23
cGAN	conditional Generative Adversarial Network , 7, 13
CIN	Conditional Instance Normalization , 12, 13
CIR	Category Image Retrieval , 16
CNN	Convolutional Neural Network , 2, 6, 8, 9, 46
COCO	Common Object in Context , 4, 11
CPMs	Convolutional Pose Machines viii, 6, 7, 9
CPN	Cascaded Pyramid Network , 8

F

FID	Fréchet Inception Distance , 15, 29
FLIC	Frames Labeled In Cinema , 4

G

GAN	Generative Adversarial Network , 7, 13, 15, 19

H

HPE	Human Pose Estimation , 2–7, 10, 38

I

IIR	Instance Image Retrieval , 16
ILP	Integer Linear Programming , 9
IN	Instance Normalization viii, 11, 12
IoU	Intersection over Union , 10
IS	Inception Score , 15, 29

L

LPIPS	Learned Perceptual Image Patch Similarity , 16, 29
LSP	Leeds Sports Pose , 3

M

MPII	Max Planck Institute for Informatics viii, 3, 4
MSE	Mean Square Error , 13, 15

N

NMS	Non-Maximum-Suppression , 8
NST	Neural Style Transfer , 2, 13, 15

O

OKS	Object Keypoint Similarity , 10

P

PAF	Part Affinity Field , 9
PAF	Part Association Fields , 9
PCk	Percentage of Correct Keypoints , 32
PCkh	Percentage of Correct Keypoints head
PCP	Percentage of Correct Parts , 32
PD	Perceptual Distance , 29
PDJ	Percentage of Detected Joints , 10
PIF	Part Intensity Fields , 9

R

ResNet	Residual Network , 7, 8
RMFAB	Royal Museums of Fine Arts of Belgium , 1, 45
RPME	Regional Multi-person Pose Estimation , 8

S

SAHR	Scale-adaptive Heatmap Regression , 10
SIFT	Scale-Invariant Feature Transform , 17

SMPL Skinned Multi-Person Linear , 3

V

VAE Variational Autoencoder , 15

W

WAHR Weight-adaptive Heatmap Regression , 10

List of Code Fragments

1

Introduction

1.1 Problem definition

To make art collections more accessible, museums put a huge effort in digitalizing their catalogue. However, they don't contain much metadata about the content and it is time-consuming to enhance them manually. To make this process easier, they want to utilize computer vision. Art collections (paintings, statues, drawings, etc.) turn out to be less interpretable by the algorithms that were developed for photography over the last few decades. These scan the images in search of recognizable objects and add their labels to the metadata. Even the latest state-of-the-art technology, struggles to recognize objects when pointed at a painting in a museum. A solution may be to start over and have paintings annotated by humans.

This has been done in 2 recent projects: Saint-George-On-A-Bike [20] and INSIGHT [21]. However, paintings are very complex while manual annotation doesn't scale and is very expensive. For example, 10,000 paintings were annotated by the Royal Museums of Fine Arts of Belgium (RMFAB) with no clear return on investment [22]. They spent a year on this and this is not something they want to repeat. How can we automate this process and ensure that state-of-the-art computer vision models give good results on paintings and artworks?

Among the different tasks that can be improved on is pose estimation. Pose estimation allows the art collection to be searchable based on different poses. This will be the focal point of improvement for thesis.

1.2 Proposed solution

There are two proposed solutions that will be explored.

A first method: the input artwork is first converted to photographic realism on which pose estimation is then executed. With this method the pre-trained models from the state-of-the-art architectures can be reused without need to do any new adjustments. This method does require the style transfer network to have a high fidelity to realism.

A second method: if the pre-trained models can't be used, it still an option to retrain one with an augmented dataset. With style transfer the images of existing datasets can be stylized and added to the datasets. This will increase the size and variance of the dataset, making it better to train on. This can increase performance on art collections as stylized images are also being trained on, but can also potentially increase the performance on photographs.

The efficacy of these methods wil be analyzed in this thesis.

2

Literature study

In order to correctly implement a solution, we need to understand the fundamentals. These consist of two research fields: Human Pose Estimation (HPE) and Neural Style Transfer (NST). The former will be used to detect poses in the art collections, but not before the latter has tried to make an improvement. Following will be an overview of the available research in these domains. Discussing what the goals of them are, how they achieve it, what their challenges are and their limitations.

2.1 Human Pose estimation

Here, HPE will be explored. The purpose and task is going to be explained with a compilation of the history of the major architectures. The different datasets used for training and evaluation metrics will be carefully analyzed.
HPE aims to detect human features from input data such as images and videos. It's an elementary part of computer vision with many applications among which are human action recognition (sign language), human tracking (surveillance), and human-computer interaction (video games). This is an extensively researched area with a diverse range of different techniques. This chapter will try to give an overview of all the many challenges and proposed solutions. The focus will be on deep learning models, which have surpassed classical solutions significantly. Specifically, around 2D monocular HPE eg., [23, 3, 24, 25].
The human body has a high degree-of-freedom due to all the limbs, self-similar parts and body types, which may cause self-occlusion or rare/complex poses. The variations in configuration are made even larger due to clothing, lighting, foreground occlusion, as well as viewing angles and truncation, among others, as shown in Fig. **??**. This makes HPE one of the most difficult tasks in computer vision [26, 2].

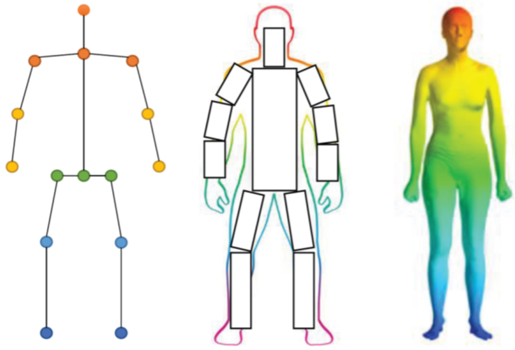
2.1.1 Representation

An important factor in HPE is how the pose will be represented. Depending on the needs of the problem you can have a skeleton-based, contour-based, or volume-based solution [2] as seen in Fig. **??**.

Skeleton-based model

The skeleton is build of a tree-structured set of keypoints that represent the joints of the human body. These can be explicitly described by their coordinates in 2D or 3D space [4]. More suitable for a Convolutional Neural Network (CNN) however is a heatmap which constructs a 2D Gaussian kernel around a keypoint [24, 27]. They are easily implemented and became the

Figure 2.1: The various challenges HPE solutions face. Images from MPII dataset. [1, 2]



(a) Skeleton (b) Contour (c) Volume

Figure 2.2: Models for pose representation [3]

dominant representation. While the skeleton-based model is a compact and flexible representation it suffers in this aspect by not being able to hold texture or shape information [3].

Contour representation

To capture the shape of the body parts, contour representation uses rectangles to estimate the body contours. These methods include cardboard models [28], which assumes people can be represented as a group of planar patches, and Active Shape Models (ASMs) [29], which tries to fit body part shapes to an image, were mainly in use in earlier HPE methods [2].

Volume representation

Volumetric geometric shapes can also be used as a method of representation. Earlier methods used simple shapes like cylinders, conics, and other shapes [30]. Volume representation is a 3D mesh that represents the human body. The most used model is Skinned Multi-Person Linear (SMPL), which includes natural pose-dependent deformations imitating soft-tissue dynamics [31].

For the purpose of our research, a simple model is the only thing we need. We only need to be aware of the most essential joints to label a pose. This makes the skeleton-based model the ideal representation to work with and will be the focus of further study.

2.1.2 Datasets

There are several publicly available datasets. There are some that are outdated and we will leave those out, focusing only on datasets used for deep learning.

1. **Leeds Sports Pose (LSP) Dataset [32]** contains 2,000 images found on Flickr using 8 different tags looking for sport activities (athletics, badminton, baseball, gymnastics, parkour, soccer, tennis, and volleyball). Each person has

14 keypoints. An extended version was later introduced [33], now consisting of 10,000 images. For this set they only focused on the more challenging tags (parkour, gymnastics, and athletics).

2. **MPII Human Pose Dataset [1]** contains 24,290 images with 40,522 labeled people. They were extracted from YouTube videos found by querying for physical activities. Each person has 16 keypoints and also includes occlusion labels.

3. **Common Object in Context (COCO) Dataset [34]** is a large-scale dataset for a wide range of computer vision algorithms. For HPE, the set contains more than 200,000 images in which 250,000 persons are annotated. Each person has 17 keypoints, a bounding-box and visibility labels. This dataset has become the most popular for benchmarking.

4. **Frames Labeled In Cinema (FLIC) Dataset [35]** contains 5,003 images extracted from Hollywood movies. They ran a person detector which collected 20,000 images from 30 movies. Occluded and difficult poses were then removed leaving only 5,000 images to be annotated. Only the upper body received 10 keypoints.

5. **AI Challenger Human Keypoint Detection (AIC-HKD) Dataset [35]** contains 300,000 images found using Internet search engines. In these, over 700,000 humans are annotated. Each person has 14 keypoints, a bounding-box, as well as visibility and left/right labels.

6. **CrowdPose Dataset [36]** puts an emphasis on crowded images. 30,000 images from MPII, glsCOCO and glsAIC-HKD were measured with a Crowd Index, which evaluates the crowdedness. Finally, 20,000 images are selected and 80,000 persons annotated. Each person has 14 keypoints and a full-body bounding box.

7. **Human-Art Dataset [37]** bridges the gap between natural and artificial images. The set contains 50,000 high-quality images with 123,000 annotated humans. Each person has 17 keypoints, bounding boxes, self-contact points, and text information.

2.1.3 Discriminative Methods and Generative Methods

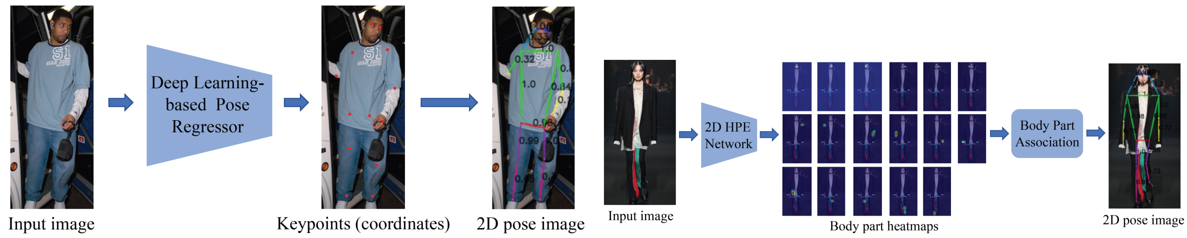
Before deep learning became prominent in HPE there were already a number of different methods in use. Some of these methods are compatible with the deep learning methods and were thus adopted. An early distinction is between generative and discriminative methods.

Generative Model

A generative method will work with prior beliefs about the pose. More information about this can be found in the section about representation 2.1.1. It will project the pose on the image and verify it with the image data. If they don't comply, the pose is adjusted using the descent direction found by minimizing an error function [38].

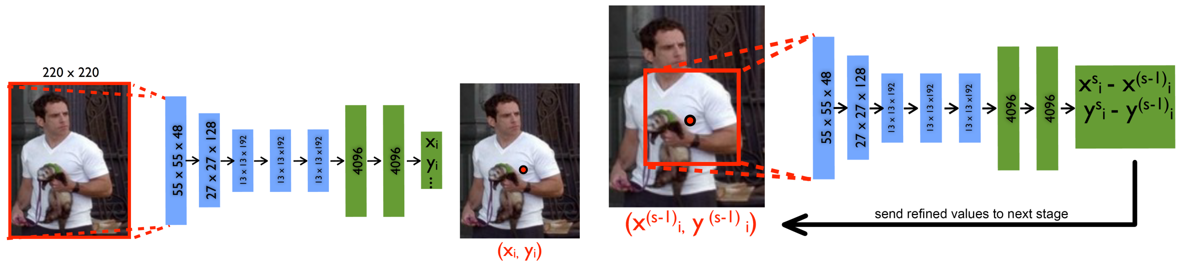
Discriminative Model

Discriminative methods on the other hand, try to map the pose on the image data with learned models. There are several methods in this category, among which are the deep learning-based methods. The deep-learning methods are further categorized by the following sections.



(a) Regression Methods (b) Heatmap-based Methods

Figure 2.3: The different methods of single-person human pose estimation.[3]



(a) Initial stage (b) Stage s

Figure 2.4: Convolution layers in blue and fully connected layers in green. The initial stage is applied to the whole images, while in stage s it will work on a sub-image based on the result of the previous stage.[4]

2.1.4 Single-Person Methods

Single-person pose estimation will try to evaluate only one pose from an image. There are 2 major methods that are in use: regression methods and detection-based methods.

Regression-based Methods

The regression-based methods learn a network that maps all the body keypoints to the image-data directly as show in 2.3a.

The first successful deep learning model came from Toshev and Svegedy [4] and is considered the switch in paradigm from classic approaches to deep learning HPE. Toshev et al. uses a 7-layered model with 5 convolution layers and 2 fully-connected layers for the pose regressor, based on AlexNet for its simple but effective architecture [39]. They then cascade the resulting found keypoints of this model to itself where it refines it using the area around the keypoints. While the network is the same, the different stages will have different learned parameters. With every stage the found keypoints become more accurate. A illustration of this can be found in Fig. 2.4.

Carreira et al. [40] introduce an Iterative Error Feedback which is a self-correcting model using top-down feedback. Using the image-data and a starting pose modeled as a heatmap, the model, based on GoogLeNet [41], will predict an error for each keypoint. The pose is then corrected based on the error and fed back into the model as a heatmap with the image. With each iteration it converges towards the solution instead of making the prediction in one go. Regression-based methods map the keypoints directly on the image, making it a non-linear problem. This will cause less robust generalization however [24].

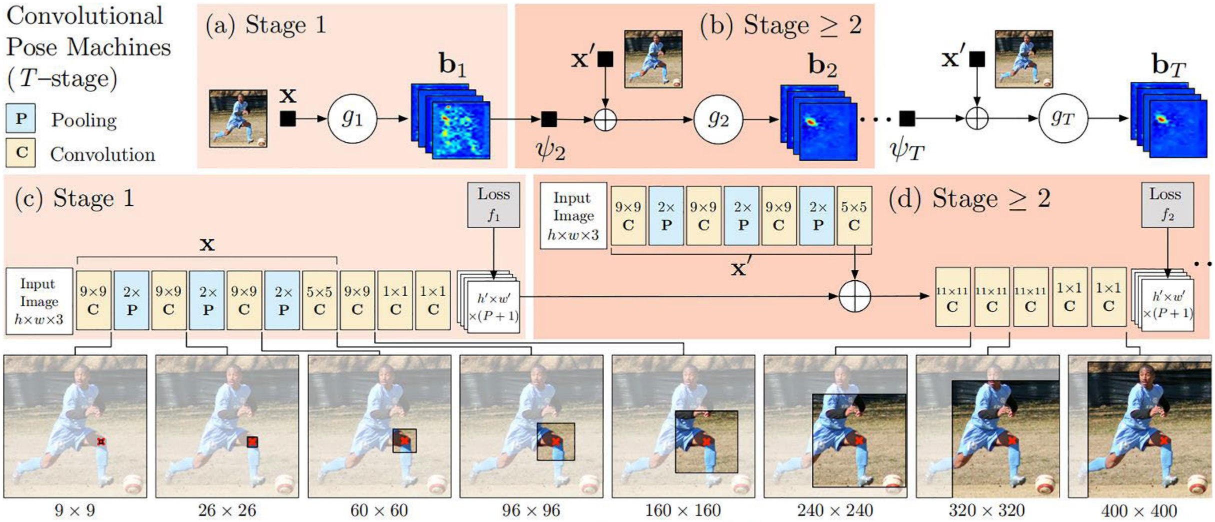
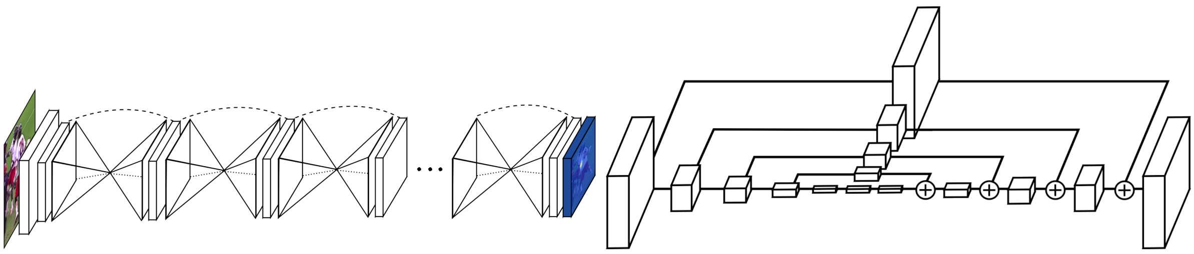


Figure 2.5: Architecture and receptive fields of CPMs. (a) and (b) represent the pose machine architecture.[5] (c) and (d) show the corresponding convolutional networks used by CPMs.[6]



(a) Stacked Hourglass

(b) Hourglass Module

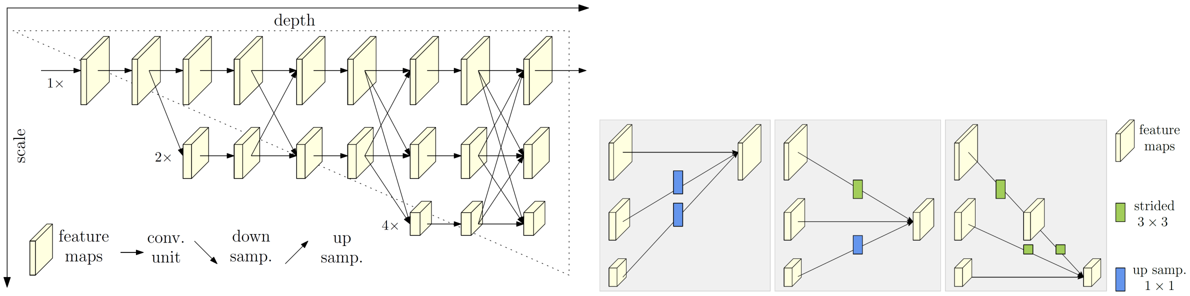
Figure 2.6: The structure of a "stacked hourglass" network and a single "hourglass" module.[7]

Heatmap/Detection-based Methods

The detection-based methods will first estimate the individual body parts using heatmaps, which leads to an easier optimization and a more robust generalization [25]. Most of the latest HPE methods use heatmaps because of this. After the joints are found they are then assembled to fit a human skeleton. This process is shown in 2.3b.

Tompson et al. [42] proposed a hybrid architecture where the detection of body parts is handled by a CNN and a Spatial-Model to bring those together. The first step produces many false-positives and these are removed in the second step by restricting joint inter-connectivity to enforce correct anatomy. They build on this in [43], where they used a cascade to refine predictions.

A fundamental work written by Wei et al. [6] combines convolution networks with Pose Machines [5]. Pose Machines is an iterative architecture which consists of 2 models: the first is used for stage 1 where it extracts potential heatmaps for the joints. The second model is used for subsequent stages where the result of the previous stage is fed in together with the results of its own convolution network on the input image. This gradually refines the predictions for the joints and their positioning. 2.5 shows this process.



(a) HRNet

(b) Multi-Scale Fusion

Figure 2.7: The architecture of the High-Resolution network and how it applies multi-scale fusion.[8]

Another influential work was being written at the same time by Newell et al. [7]. Similar to CPMs, this is also an iterative architecture. They suggest what they call a "stacked hourglass" network, where "hourglass" modules are repeated 2.6a. In an "hourglass" module, first, the features are downsampled and afterwards upsampled again 2.6b. This network captures different spatial relationships between joints at different resolutions. Several other works [44**?** , 45] have since improved on the network design.

Both these use intermediate supervision to tackle the problem of vanishing gradients. This still doesn't build a deep sub-network for feature extraction which limits the estimations. This has become less of a problem with the emergence of Residual Network (ResNet) [46] which allows better back-propagation at deeper levels through shortcuts.

A more recent work by Sun et al. [8] maintains the high-resolution representations instead of working the high-resolution from the low-to-high sub-network. After a first high-resolution sub-network, it gradually adds high-to-low sub-networks in parallel to predict multi-resolution features. Before each branch, they apply multi-scale fusion, which joins the predicted features from each scale on each scale. Both are shown in 2.7. This network has proven very effective and inspired several variations [47, 48, 49].

With the emergence of neural networks also came Generative Adversarial Networks (GANs) [50], which proved useful for HPE. They are employed to improve constraints of joint inter-connectivity and infer occluded body parts.

Chen et al. [9] propose a structure-aware convolution network using a stacked hourglass as generator which generates heatmaps for each joint. They use 2 discriminators, one to discriminate between low- and high-confidence predictions, another for real and fake poses. The network is designed as a conditional Generative Adversarial Network (cGAN) [51], which allows it to generate pose heatmaps as well as occlusion heatmaps.

A more classic GAN is used by Chou et al. [52], where they use a stacked hourglass network for both the generator as the discriminator. The generator predicts the heatmaps for each joint and the discriminator distinguished between the real and fake ones.

2.1.5 Multi-Person Methods

With multi-person methods comes an extra layer of difficulty: they need to be able to detect each person separately. To solve this problem multi-person methods propose several solutions. The 2 most popular are top-down and bottom-up methods.

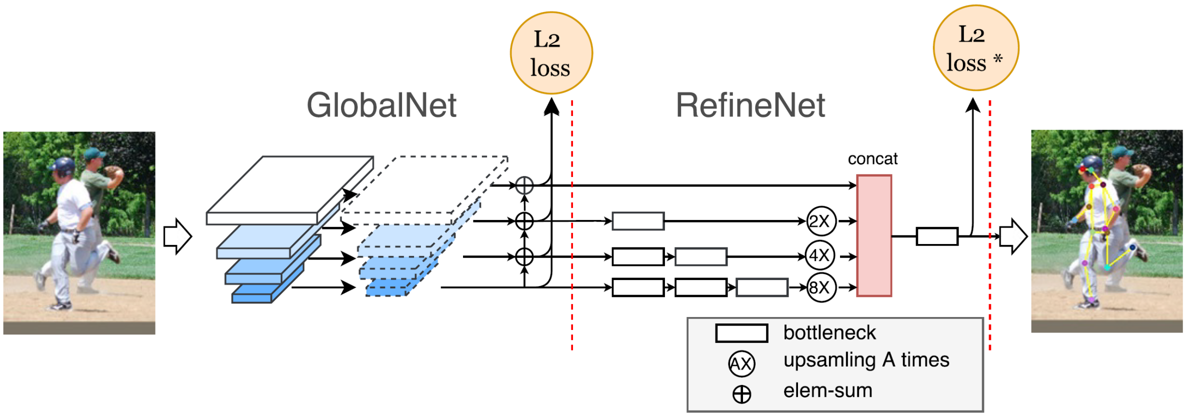


Figure 2.8: Cascaded Pyramid Network. "L2 loss*" means L2 loss with online hard keypoints mining.[9]

Top-Down Methods

This method will first try to detect all persons in the image with a human detector. Each person is cropped by the bounding box and a single-person estimator predicts a pose for each person.

Occlusion and truncation are a regular occurrence in multi-person scenes and inevitable problem. One of the early multi-person models, by Iqbal et al. [53], works towards creating a robust model against occlusion. It uses Faster RCCN [54] to detect the human boundaries. After which, it applies integer linear programming for each person's fully connected graph. This technique is similar to [55], but instead of working on all globally found joints it only considers local joints. It can also handle any kind of occlusion or truncation.

The use of a human detector comes with its own sort of problems. Fang et al. [56], with Regional Multi-person Pose Estimation (RPME), try to remedy these with 2 components: They try to tackle inaccurate bounding boxes with Symmetric Spatial Transformer Network, redundant detections with Parametric Pose Non-Maximum-Suppresion. They also propose a 3rd component, Pose-Guided Proposals Generator, which can augment training samples.

Papandreou et al. [57] use a 2 stage pipeline. In the first stage, they employ the Faster RCNN detector [54]. In the second stage, they estimate the pose in each found bounding box using their own network. It predicts heatmaps using a fully convolutional ResNet and use their own novel aggregation procedure. Afterwards, they do post-processing using keypoint-based Non-Maximum-Suppression (NMS) a method of their own making.

A continuous effort is taken by Chen et al. [9] to deal with occlusion and truncation. They suggest a 2 stage architecture, a Cascaded Pyramid Network (CPN) as seen in 2.8, where first the "simple" keypoints are captured with GlobalNet, a feature pyramid network based on [58], and the "hard" keypoints are handled by their RefineNet, based on the upsampling and concatenating of HyperNet [59] and using an adapted stacked hourglass. They achieved great results and several others improved on their work [60, 61].

In more recent research, a new method was become more powerful than CNNs. The Transformer [**?**], based on attention mechanisms which are used to optimize recurrent networks [62], eliminates the use of recurrent layers, keeping only the attention mechanisms. Yang et al. [63] use this architecture because allows for better understanding of the spatial dependencies and learns at a higher rate.

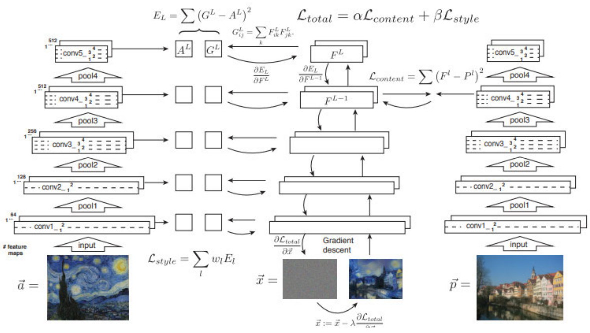


Figure 2.9: Style transfer algorithm. (Gatys et al. [10]).

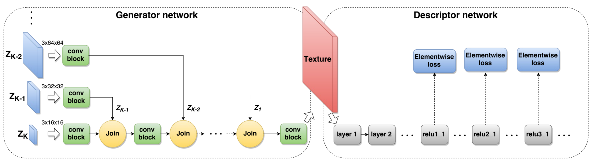


Figure 2.10: A texture network by Ulyanov et al. [11]. The generator network (left) is the only one that changes. The descriptor network (right) is used to calculate the loss.

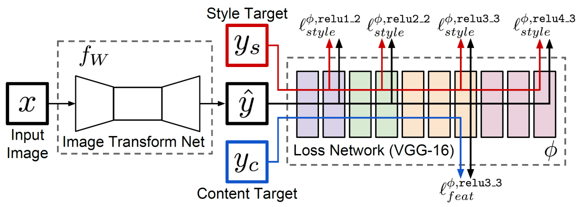


Figure 2.11: An image transformation network by Johnson et al. [12]. The image transform network (left) is the only one that changes. A loss network (right) is used to define perceptual loss functions.

Bottom-Up Methods

A different approach is taken with bottom-up methods. They first locate all joints in the image and then assemble them in potential humans.

DeepCut by Pishchulin et al. [55], one of the first multi-person models using CNNs. Using Fast R-CNN [54], it detects the body parts and labels each. With the joints found, it then uses Integer Linear Programming (ILP) to assemble them. This method is very computationally expensive; NP-hard. Insafutdinov et al. [64] therefor introduce a stronger part detector and better optimization strategy with DeeperCut.

CPMs make a return with OpenPose by Cao et al. [65], they're used to predict the joints with heatmaps and Part Affinity Fields (PAFs). A part affinity field also encodes the position and orientation of the limb which makes the assembly of joints into different poses possible. They can achieve real-time results with this method, and several others have improved on their design [66, 67, 61]. The high performance is only applicable to high-resolution images. Low-resolution images or images with occlusions perform poorly.

Kreiss et al. [68] continue on the idea of fields and introduce the Part Intensity Fields (PIF) and Part Association Fields (PAF). First, they predict the location of the different joints with PIF. Afterwards, they use PAF to find the inter-joint relationships. They are able to outperform any previous OpenPose-based proposals on low-resolution and occlusions.

Newell et al. [69] introduce a new method called associative embedding for supervising CNNs both detection and grouping. This is a single-stage architecture as opposed to the two-staged architectures previously discussed. They make use of the stacked hourglass network from [7] with some small modifications.

Continuing on the idea of associative embedding, Cheng et al. [47] use HRNet [8] as backbone for their HigherHRNet.

Their method focuses on the scale-variance problem; a problem which hasn't been studied much, so it can localize keypoints for small persons better. Lou et al. [70] introduce Scale-adaptive Heatmap Regression (SAHR) and Weight-adaptive Heatmap Regression (WAHR) to the scale-variance problem. SAHR adaptively adjusts the standard deviation of each heatmap corresponding with the scale of the person. WAHR rebalance the foreground and background samples, so SAHR can work to its fullest extent.

Summary

An important challenge for HPE is making predictions in scenes with hight occlusions. Top-down models achieve state-of-the art performance in almost all benchmark datasets [2]. Top-down models has difficulty with overlapping bodies and human detectors might fail finding humans there. To the same extent, bottom-up models will have greater inaccuracy with grouping in occluded scenes. Computationally, the top-down model's speed is limited by the number of people found. The higher efficiency of bottom-up models, make them more suitable for real-time applications.

2.1.6 Evaluation Metric

The evaluation of an HPE looks to measure the accuracy of the location of predicted joints. Because of the different number of features and tasks across datasets, there are also several different evaluation metrics in use. Explained next will be the most commonly used metrics.

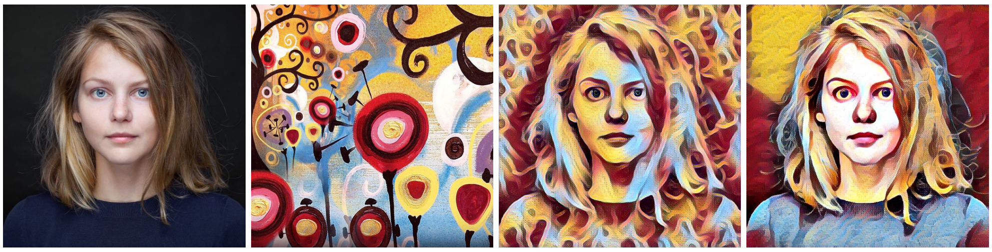
1. **Percentage of Correct Parts (PCP)**, proposed by Ferrari at al. [71] measures the detection rate of limbs. A limb is considered the area between 2 joints and viewed as detected when the distance between the predicted joints and the real joints is less than halve the length of the limb. This method penalizes shorter limbs and to address this, Percentage of Detected Joints (PDJ) was introduced which instead measures it with a fraction of the torso diameter. The higher, the better.

2. **Percentage of Correct Keypoints (PCK)**, suggested by Yang et al. [72] measures the accuracy of the predicted keypoints. The keypoints should be within a certain threshold which is a fraction of the person's bounding box size; denoted as PCK@0.2 when it should be less than 20%. It can also be 50% of the head's length; denoted as PCKh@0.5, which makes it "articulation independent". The higher, the better.

3. **Average Precision/Recall (AP/AR)**, is measured by Yang et al. [72] by counting a keypoint that is within a certain threshold of the ground truth as a true positive. For Lin et al. [34], the AP is calculated by measuring the Object Keypoint Similarity (OKS) which is similar to Intersection over Union (IoU) in Object Detection. The OKS is defined as:

$$\text{OKS} = \frac{\sum_i \text{exp}(-d_i^2/2s^2k_i^2)\delta(v_i > 0)}{\sum_i \delta(v_i > 0)} \tag{2.1}$$

   Here, d_i is the distance between the predicted keypoint and the ground truth. The distance is run through a unnormalized Gaussian with a standard deviation of sk_i which yields a similarity that ranges between 0 and 1. s is the scale, calculated as the root of the segment area, and k_i is a constant for each keypoint that controls falloff. OKS is the mean of visible keypoints ($v_i > 0$). These can be used to calculate Average Precision (AP) and Average Recall (AR) at different thresholds. 10 different metrics are used to calculate the performance of a model: AP$^{0.5}$ (where the



(a) Content Image (b) Style Image (c) StyleNet with BN (d) StyleNet with IN

Figure 2.12: A comparison between (c) BN and (d) IN.[13]

OKS threshold is 0.5), AP$^{0.75}$ and AP (the mean of 10 values from OKS $= 0.50$ to 0.95 with a 0.05 step), as well as, AP^{M} for medium scaled objects and AP^{L} for large scaled objects. The same are calculated for AR. The higher, the better.

2.2 Image Style Transfer

Image Style Transfer is the technique of applying the style of one image to the content of another. Classically this was a problem reserved for only artists, but more recently this has also interested computer scientists. There are several different ideas on how this can be achieved, ranging from how to separate the style from the content, to how well an algorithm can generalize. An overview of all the different challenges and solutions will be given in this chapter.

2.2.1 Datasets

Due to a lack of benchmark datasets, multiple papers will mix and match from different datasets, like COCO or ImageNet [73].

1. **Cityscape Dataset [74]** consists of 2975 images of cityscapes with semantic annotations.
2. **Facades Dataset [75]** consists of 400 images of building facades with architectural annotations.
3. **Maps Dataset [76]** consists of 1096 images of maps and areal photos gathered from Google Maps around New York City.
4. **Edges2shoes Dataset [77]** consists of 50,000 paired images between edges and photos of shoes.
5. **Edges2handbags Dataset [78]** consists of 137,000 paired images between edges and photos of handbags.
6. **Horse $\leftrightarrow$ Zebra [66]** consists of 2,500 images of 512x512 horses and zebras, that were sampled from ImageNet [73].
7. **Animal Face High Quality [79]** consists of 15,000 high quality images of 512x512 animal faces, including cat, dog and wildlife.

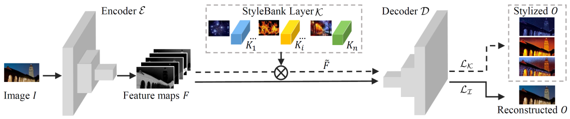


Figure 2.13: The stylebank network by Chen et al. [9].

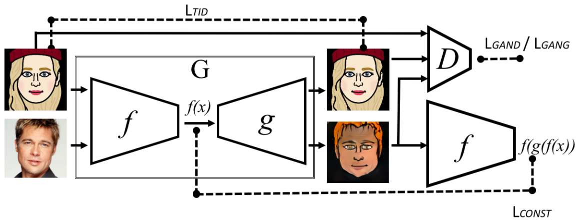


Figure 2.14: The domain transfer network by Taigman et al. [14].

8. **Night2Day Dataset [80]** consists of 20,000 images taken from time-lapse datasets and annotated through crowd-sourcing.

9. **WikiArt Dataset [17]** consists of 80,000 fine-art paintings. All are annotated for 27 styles, 60,000 are annotated for 20 genres and 20,000 for 23 artists.

2.2.2 Optimization-based Networks

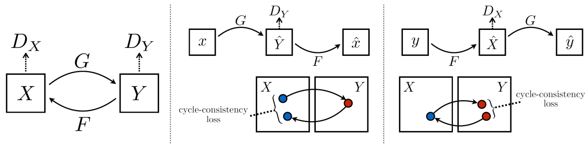
Gatys et al. [10] introduce deep neural networks to image style transfer. Using a modified VGG-network [81], they extract the features of an image by reconstructing the content from the feature maps in the higher layers on a white noise image. The same is done for the style of the other image. It extracts the style representation of the image by using the Gram matrix to represent style features of the image and then reconstructs it on the same white noise image. The Gram matrix is the vector product of two sets of vectorized feature maps. This method is shown in 2.9. They remark that the resolution of the images affects the performance of the algorithm and is thus restricted to low resolutions. At the same time, the synthesized images contain some low-level noise, but this can possibly be removed with a denoiser.

2.2.3 Feed-forward Generation Networks

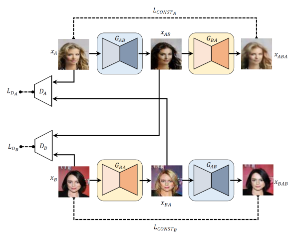
To improve the performance, Ulyanov et al. [11] suggest the use of a feed-forward generation network instead of back-propagation Backpropagation requires an iterative process to change the pixel values to match the desired statistics. A feed-forward network can do this in a single evaluation. To train such a network they use a pre-trained network for image classification, and calculate a texture and content loss like [10], as shown in 2.9. Johnson et al. [12] propose a very similar method as can be seen in 2.9

Since their contribution did increase the speed, but at the expense of quality, Ulyanov et al. [13] suggest further improvements to their network. First, they replace BN [82] with IN which alone has a significant impact on quality as can be seen in 2.12. Second, they learn the generator to sample from the Julesz ensemble [83] which improves variation in the outputs.

Dumoulin et al. [84] note that previous feed-forward networks are limited to one style. In order to facilitate many different styles, there would need to be a network trained separately for each which limits the applications for mobile devices. In order to make the network more memory efficient, they propose a conditional style transfer network; given a content image and a style name, it transforms the image to the corresponding style. They argue that after normalization each style can be distinguished by specializing scaling and shifting parameters. They call this Conditional Instance Normalization

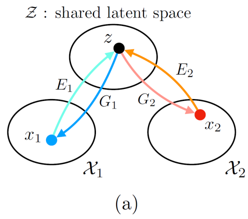


(a) As illustrated by Zhu et al. [66].

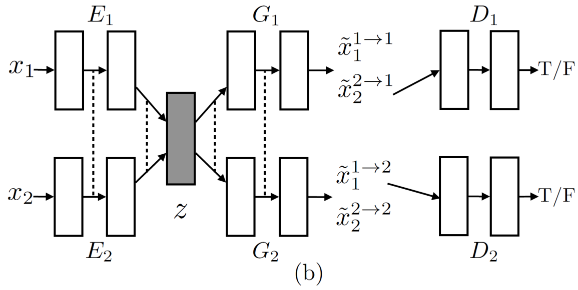


(b) As illustrated by Kim et al. [62].

Figure 2.15: The cycle-consistent network.



(a) The shared latent space assumption.



(b) The unsupervised image-to-image translation network.

Figure 2.16: Liu et al. [15].

(CIN). Since it only changes the scale and shift parameters for different styles, the network requires fewer parameters. Of the 1.6M parameters, only 3K are needed for the different styles.

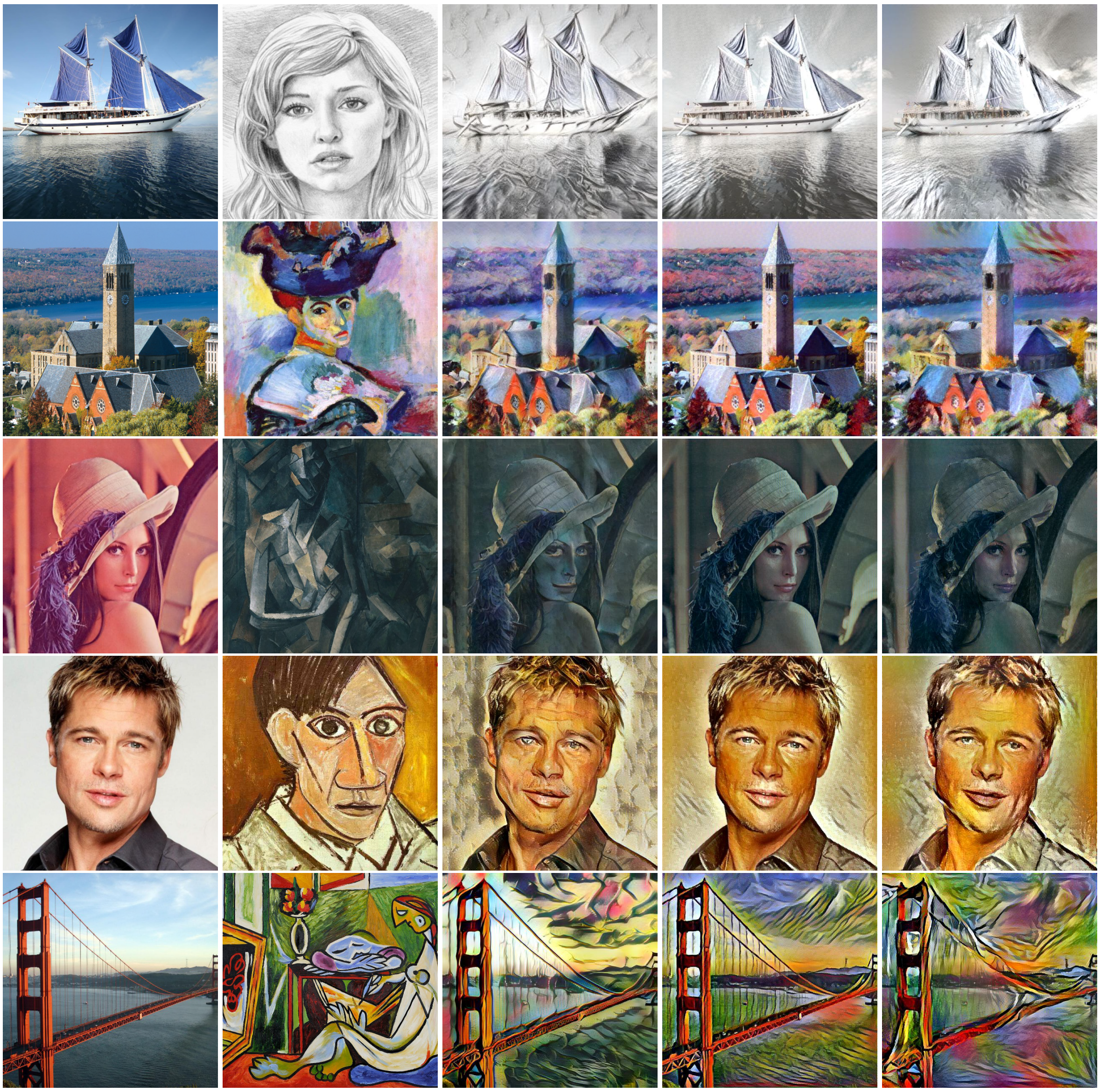
Another network that puts a focus on multiple styles comes from Chen et al. [9]. They propose a StyleBank, as seen in Fig. **??**, which can store multiple convolution filter banks each representing a different style. They use an auto-encoder network with in between a StyleBank layer. During training, for each $T+1$ iterations the entire network is first trained with a perception loss for the first T iterations. Then only the auto-encoder network is trained with a Mean Square Error (MSE) loss. This way the auto-encoder only retains the content and the StyleBank layer only the different styles. This also allows to lock the encoder and decoder to learn a new style afterwards.

While CIN allows for multiple styles, it's still limited to the ones that were seen during training. Huang et al. [85] try to remedy this by introducing an Adaptive Instance Normalization (AdaIN) layer. Unlike the other normalization techniques, AdaIN does not have affine parameters, and will adaptively compute these from the style image. 2.17 shows how well the different networks can handle unseen styles.

2.2.4 Generative Adversarial Networks

With the introduction of GANs, the quality of generative models have greatly increased. It is not surprising then that this got picked up in research for NST.

Among the first was Isola et al. [76] who use a cGAN. With cGAN, the generator network has an extra input which here is the image to be translated. They use the network from [86] which uses modules of the form convolution-BatchNorm-ReLu[82]. Additionally, in order to pass shared features in the generator they add skip connections like with "U-Net" [87]. For the discriminator, which they call PatchGAN, they validate $N \times N$ patches and take the average as output. They take this loss together with the $L1$ loss because $L2$ loss produces blurry results.



(a) Content Image (b) Style Image (c) Huang et al. (d) Ulyanov et al. (e) Gatys et al.

Figure 2.17: A comparison between different style transfers where the style was not seen during training.

This still requires paired training samples, while Taigman et al. [14] are doing research in unsupervised domain transfer. Domain transfer can be used for NST, but this is not possible the other way around. Their network uses a encoder-decoder as the generator and they assume that $f(x)$ is constant between 2 domains. The discriminator has a ternary output and distinguishes between real, fake and reconstruction. They add several new loss functions which check the consistency between the 2 domains (consistency loss) and if G performs perfect reconstruction (reconstruction loss). This can be seen in **??**. For f, they use a pre-trained network that is trained on paired samples.

In order to make the network completely unsupervised, Yi et al.[88] propose DualGAN, Kim et al. [62] DiscoGAN and Zhu et al. [66] CycleGAN, which are all 3 essentially the same proposal. The entire model consists of 2 cycle-consistent networks where each translates from one domain to the other. A cycle-consistent network will first translate the input to target domain and then back to the original domain. Each domain has a discriminator which compares the real input from one network with the fake from the other; the adversarial loss. As seen in 2.15b. In addition to this there's a cycle-consistency loss, which is the MSE between the input and the reconstructed image as you can see in 2.15a. The goal is to minimize the adversarial and cycle-consistency losses, while maximizing the discriminators' accuracy. Zhu et al. [66] also introduce an identity loss.

Liu et al. [15] introduce the latent space concept which assumes that paired images from different domains can be mapped to a shared latent space with the same latent representation. The network consists of 2 domain image encoders E_1 and E_2, 2 domain image generators G_1 and G_2, and 2 domain discriminators D_1 and D_2. As can be seen in **??**. The encoders and generators are paired and form a Variational Autoencoder (VAE) [89]. The encoder maps the input to latent space, and the generator reconstructs the image. This is the reconstruction loss. They use weight-sharing, which shares the weight of the last 2 layers of the encoders and of the first 2 layers of the generators. The generators and discriminators are paired to form a GAN. The generator can also construct an image from the latent code from the other encoder's input. This image is used to train the GAN. They also show that the shared-latent space assumption implies cycle-consistency, which is the final loss function of the network.

2.2.5 Evaluation Metric

There are several methods of evaluating the quality of a generated image. A first metric was through human evaluation; a score was given based on generation quality. This proved to be inconsistent as a person's perception can change over time. Afterwards, new metrics were introduced which will be discussed here. [90]

1. **Perceptual Distance (PD)** is proposed by Johnson et al. [12]. It uses the VGG-16 network [81] trained on ImageNet [73] to define perceptual loss functions. These are extracted from the layers for the style and content images, and compared to the generated image. The lower the score, the better.

2. **Inception score (IS)**, as described by Salimans et al. [91], uses a pre-trained Inception model [92] to describe the quality of the generated images. It prescribes that the entropy of the distribution of predicted labels for individual images needs to be minimized while the entropy of the distribution across all images need to be high. This equates to each image having generated a distinct label and the labels being equally distributed. The closer to 1, the better.

3. **Fréchet Inception Distance (FID)** is the most used measurement and suggested by Heusel et al. [93] to enhance the Inception Score (IS). IS is only calculated on the distribution of the generated images. Fréchet Inception Distance

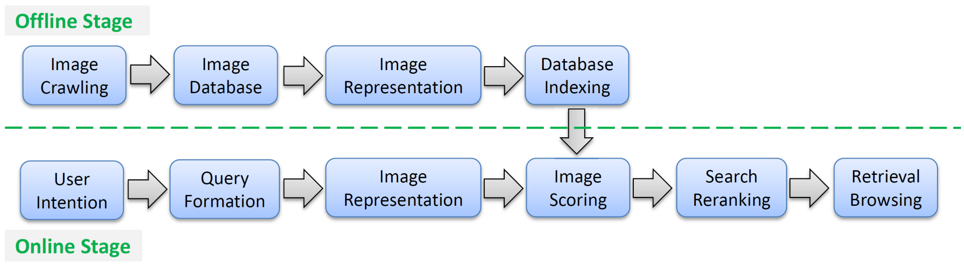


Figure 2.18: The general workflow of CBIR. [16]

(FID) uses the distribution of both real and generated images. It calculates the Fréchet distance [94] between the Gaussian distributions of real and generated images. The Gaussians are formed from the coding layer of the Inception network [92]. The lower, the better.

4. **Learned Perceptual Image Patch Similarity (LPIPS)** is a metric developed by Zhang et al. [95] and the second most popular. It calculates the distance between the activations of the hidden layers in an object detection model (several models are proposed). They show that this correlates closely to human perception. It can also be used to evaluate the diversity of a network by calculating the average Learned Perceptual Image Patch Similarity (LPIPS) score of a pair of randomly generated output. The higher, the better.

Summary

There are plenty of other evaluation metrics available that also try to correlate closely to human evaluation, but they are mostly just attempts to improve previously discussed metrics. Until this day, image similarity metrics continue to be a challenging problem.

2.3 Content Based Image Retrieval

CBIR, a long-established research area, is the task of finding semantically matched or similar content images for a specified query image. This has become increasingly relevant with the exponential growth of image and video data and the need to effectively search these image collections. specifically, CBIR has been used for person re-identification, remote sensing, medical image search, and shopping recommendation in online marketplaces, among many others [18]. Image retrieval can be categorized into 2 different groups: Category Image Retrieval (CIR) and Instance Image Retrieval (IIR). CIR's goal is to find images within the same category as the query, while IIR tries to find images with a particular instance given in the query image. The general workflow of CBIR is illustrated in **??**. This paper will only discuss query formation, image representation, image scoring, and search re-ranking.

2.3.1 Query Formation

There are several ways that a query can be formatted. A user might want to find images based on keywords which is your standard classification task. Instead of just giving a series of keywords, these can also be arranged in a layout. A query by concept layout will then search for an image with the same arrangement [96]. Similarly, a query by color layout will search

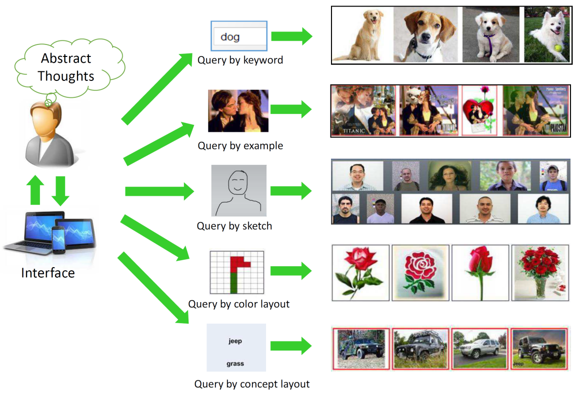


Figure 2.19: An overview of the different kinds of queries with corresponding retrieval results. [16]

for that arrangement of colors in the images [97]. It's also possible that a user wants to find images similar to a sketch (query by sketch) [98] or another image (query by example) [99]. An overview can be found in fig. 2.19. This paper will focus on query by example.

2.3.2 Image Representation

A major challenge with image retrieval is how to proficiently measure similarity between images. Clearly, directly comparing pixels values is impracticable, so methods that extract visual features from images are used. They are transformed into a fixed-sized vector which form a representation of the image. Before deep learning, hand crafted feature algorithms were used. From these, Scale-Invariant Feature Transform (SIFT) [100] was the most popular. This is still not enough for an efficient query response and visual features need to be further compressed for indexing. (Talk about codebooks some more) (See 3.1 for reason of inclusion)

2.4 Related Papers

(dirty version) Discuss following papers: Improving Object Detection in Art Images Using Only Style Transfer [101]
Enhancing Human Pose Estimation in Ancient Vase Paintings via Perceptually-grounded Style Transfer Learning [102]
Linking Art through Human Poses [103]

3

Establishing a Baseline

This chapter will establish the baseline that will be used to compare the results of the experiments with. For this, 3 algorithms for style transfer and 2 algorithms from both pose estimation will be explored. The motivation for the choices of the algorithms will be explained in full detail. The focus will be on quality instead of speed. For style transfer, the different models will be trained on 3 datasets of different art movements and evaluated based on different metrics. The precision of pre-trained pose estimation models will be measured on the COCO dataset to establish a ground truth. Afterwards, the pre-trained models will be validated on the Human-Art dataset and the stylized COCO dataset. The results of that will give an indication of how well pose estimation will work on art collections and where there's room for improvement.

There are several considerations to be made when choosing the right model. For the purpose of this thesis, the focus will be mainly on papers that have code readily available. The model must also be compatible with the preferred dataset. For style transfer, this is not a problem as the input for all models is only an image. On the other hand, pose estimation has several datasets with different amounts of keypoints, bounding boxes or other metadata. The most popular dataset and supported by most models will be uses, which is the COCO dataset. The main aspect of the the problem is quality and speed. When setting up a database for querying, there needs to be qualitative results to search through. The search itself should be fast, but this is not the subject of this thesis. At the same time, there should be a wide variation in architecture. It makes little sense to analyze two similar architectures here as that has already been explored in the corresponding papers and is not threading new ground. All these criteria are considered in the next sections as well as those uniquely for each section.

3.1 Training Style Transfer

3.1.1 Choice of Model

The most important criteria for style transfer is the quality. For the baseline, the photographs need to be inseparable of any artworks for the measurements to be useful. Pose estimation is trained on photographs, so style transfer needs to create accurate photographs for it to live up to its claimed performance. However, measuring the quality of an image is a difficult task. There are numerous metrics each based on different criteria due to the absence of a universally agreed-upon metric. [104] There is a general consensus that it should closely resemble human evaluation. Of all the different models, the more recent models aim more on finding a transformation mapping rather than merely doing a texture transfer. To keep the complexity low, this thesis will only focus on the latter, while keeping to the main advancements. As previously mentioned, a wide variation of architectures should be selected.

For this reason, AdaIN [85] was selected from the feed-forward generation networks. It's also one of the networks which can transform from an arbitrary style unlike the other selected networks. CycleGAN [66] is a major breakthrough in the training scheme of GANs and cycle loss has since been incorporated in most new models. It also has several pre-trained networks in the styles of several important artists. Another interesting concept, is that of latent space where the assumption is that there exists a common space that can encode information from several domains. [15] This is used in StarGANv2 [79] to implement a model that can transform images between several different domains using the same network. StarGANv2 will be described as StarGAN in the future. It is these models that will be analyzed. Each representing a significant contribution to the field of image-to-image translation.

3.1.2 Creation of datasets

For training, there seems to be only one good dataset that can be used and that is the WikiArt dataset. It categorized the artworks into several art movements, but also multiple genres as seen in Table 3.1. Since the transformation between styles should be as little as possible, the only useful styles here are those with high realism, however, they should not be hyperrealistic. There are plenty of styles that are compatible with these criteria and also have plenty of images to create a well sized subset. The choice of style beyond that point is completely the result of the bias of the author. This results in the selection being: Baroque, Renaissance and Impressionism. The impressionist style is chosen because it is more colorful and abstract than the others. Baroque and Renaissance are both very dark and very similar in style, but renaissance artworks are just a bit more stylized. This was a deliberate choice to see if there's possibly a difference between these attributes. The Cezanne2photo dataset [66] was used to get an idea of how big these should at least be above 500 images. A bigger dataset is better, but there are only so many artworks available. This means that the size for all except one are around 800 images. More details are given in Table **??**. When looking at the datasets mainly used by the unsupervised image-to-image models, there is a very specific focus on certain domains. AFHQ used for StarGANv2 or Horse$\leftrightarrow$Zebra show that the training images put the subject central in the image. This means that for each movement, a subset needs to be created with images that contain full body poses as well as crowded images, as this is what the pose estimation models are trained on. While there's a high variation of genres in the WikiArt dataset, they do not adequately subdivide the dataset for this problem. At first glaze, it seems that the genres "nude painting" and "portrait" would give a good set of images to use, however, there are still multiple problems. The portraits are mostly zoomed in from the chest up. There should be a higher variation in poses than that. Like with the nude paintings, but those don't have as many images to create a dataset from. Another genre that might be promising, is "genre painting", but those don't always have the model central to the image. Overall, there is still a high variety of style even within the different art movements. There is also the presence of sketches or graphite drawings. Various examples of these shortcomings are illustrated in figures 3.1, 3.2, 3.3 and 3.4.

As discussed previously, the art movements were deliberately chosen to see if certain attributes, e.g. color and abstraction, have an influence on the performance of style transfer. It is important then to have a consistent style in each dataset which is not possible to create with just splitting the genres provided by WikiArt. At the same time, there is also a need for a subset from the COCO dataset with a consistent style and the human central to the image. To achieve this, an algorithm was sought to find similar images.

Feature extraction First, an algorithm that extracts features using the VGG16 from the images was used [105]. It calculates the cosine distance between the image features, and groups them using DBSCAN [106]. This did not yield any promising

Figure 3.1: Portraits are mainly from the chest up.

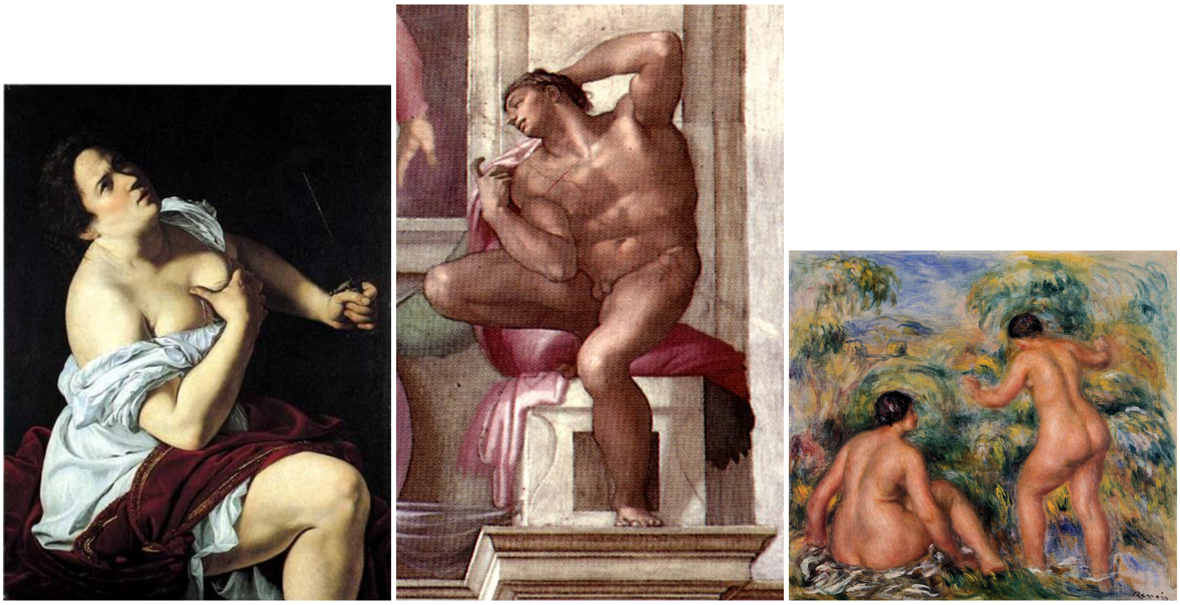


Figure 3.2: Nudes have a better variation of poses, but a small number of images. 23 for baroque, 247 for impressionism and 21 for renaissance.

Figure 3.3: In genre paintings humans are less central to the painting.

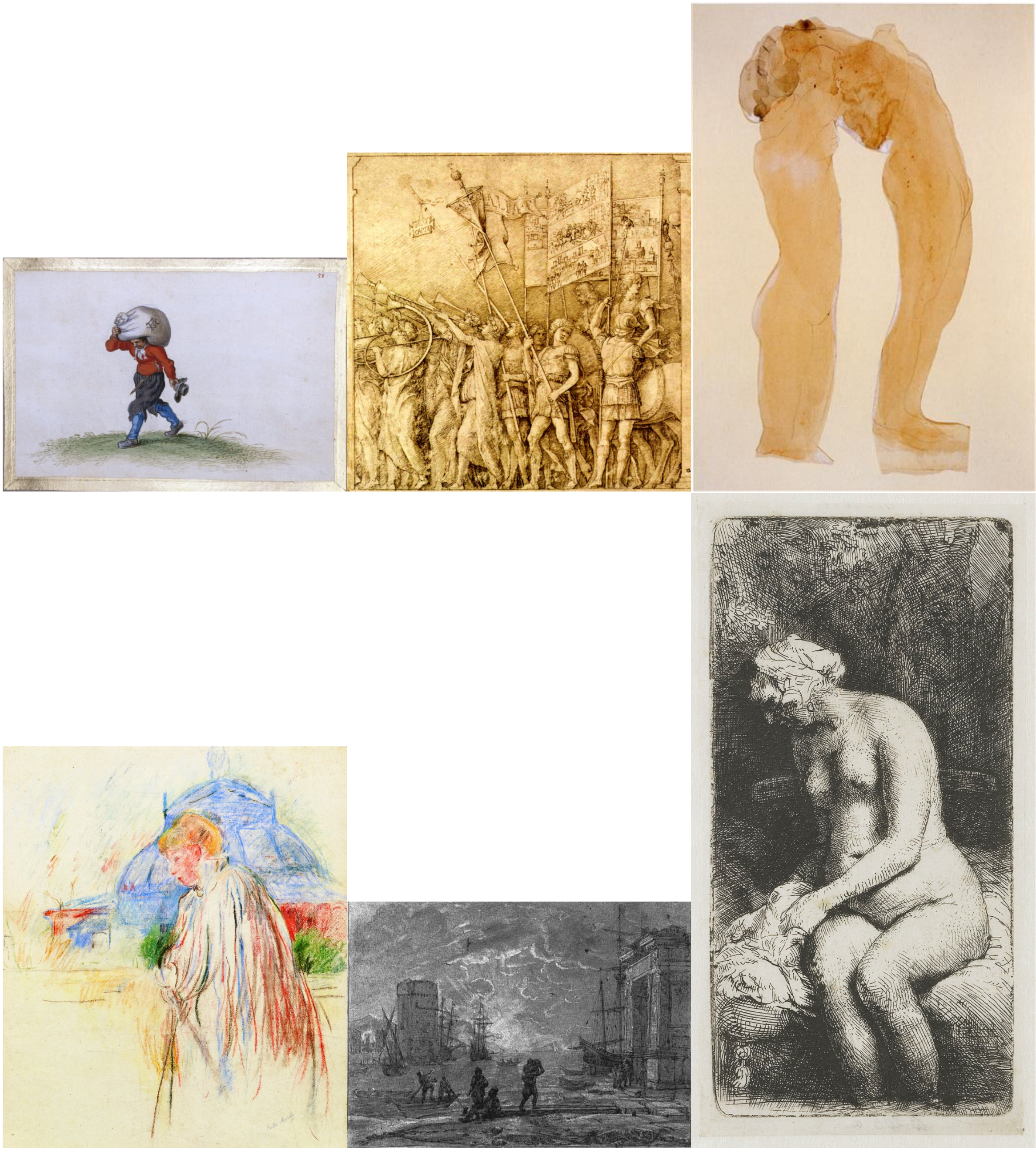


Figure 3.4: The variation in style withing the different art movements.

Figure 3.5: Example of failed query for CBIR where the flower pattern is isolated. The left image is the query image.

Figure 3.6: Example of failed query for CBIR where the car and concrete are isolated. The left image is the query image.

result. Instead of VGG16, YOLOv8 [107] was substituted for feature extraction, but this did also not provide satisfactory results.

Content Based Image Retrieval Another way to find similar images is with CBIR. Using a query image it can find similar looking images. Because this algorithm is trained to recognize similar instances and not a specific style or genre, the query image needs to be carefully selected. When there's another recognizable instance besides a person in the image it will also score images with that instance highly. Figures 3.7, 3.5 and 3.6 show how a car, a flower pattern or even just a kitchen is enough to find different instances. On the other hand, some activities are so distinct that only instances of that activity are found. The figures 3.8, 3.9, 3.11 and 3.10 shows the query images used to construct the different datasets along with a selection of the dataset.

Figure 3.7: Example of failed query for CBIR where the kitchen is isolated. The left image is the query image.

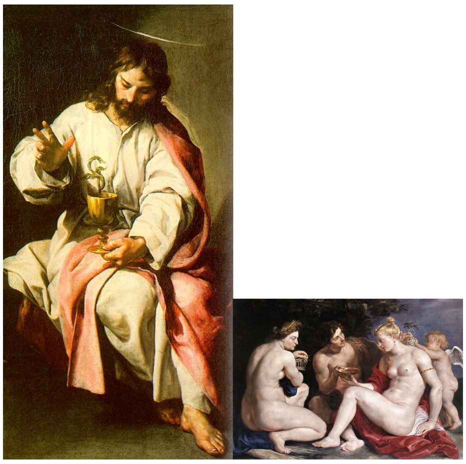
Table 3.1: List of the selected genres and names of the styles in the WikiArt dataset. [17]

Task Name	**List of Members**
Genre	abstract painting, cityscape, genre painting, illustration, landscape, nude painting, portrait, religious painting, sketch and study, still life
Style	Abstract Expressionism, Action Painting, Analytical Cubism, Art Nouveau-Modern Art, Baroque, Color Field Painting, Contemporary Realism, Cubism, Early Renaissance, Expressionism, Fauvism, High Renaissance, Impressionism, Mannerism-Late-renaissance, Minimalism, Primitivism- Naive Art, New Realism, Northern Renaissance, Pointillism, Pop Art, Post Impressionism, Realism, Rococo, Romanticism, Symbolism, Synthetic Cubism, Ukiyo-e

(a) Query images

(b) Resulting dataset

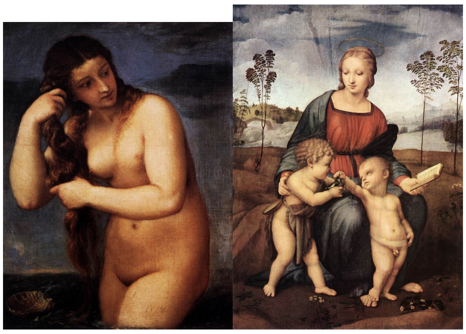
Figure 3.8: The photograph dataset consists of 825 images.



(a) Query images

(b) Resulting dataset

Figure 3.9: The baroque dataset consists of 518 images.

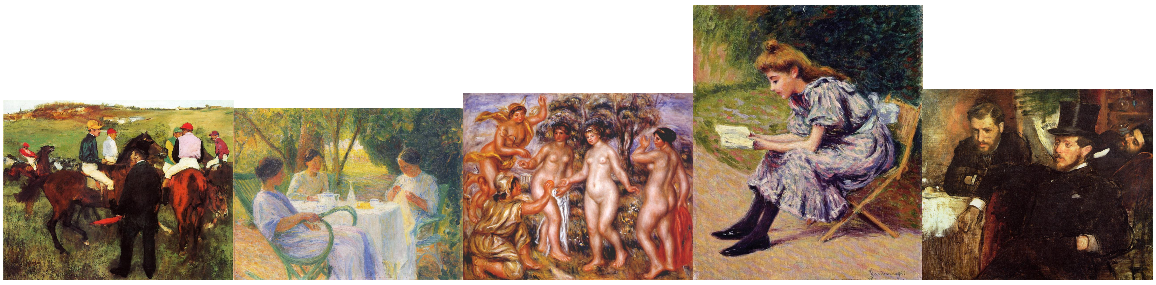


(a) Query images

(b) Resulting dataset

Figure 3.10: The renaissance dataset consists of 790 images.

(a) Query images



(b) Resulting dataset

Figure 3.11: The impressionism dataset consists of 780 images.

3.1.3 Training

From the selected models there are only 2 that require training, CycleGAN and StarGAN. AdaIN alleges that it can use any arbitrary style from a content image to do style transfer. This eliminates the need to train a new model for it and the pre-trained model can be used for the experiments. The other models will be trained with the provided default parameters. No hyperparameter tuning will be done as the goal is to measure the performance between different approaches and not optimize a single model.

CycleGAN was trained using a different number of epochs for each style to compare the performance . Baroque was trained for 200 and 2000 epochs, renaissance for 500 epochs and impressionism for 750 epochs.

StarGAN does not use epochs to determine the training progression, or, at least, the pytorch implementation doesn't. The model was trained to find a mapping between all different datasets for 100,000 iterations.

3.1.4 Results

Qualitative Evaluation

As shown in Fig. 3.12, AdaIN removes more of the details of the content than CycleGAN does, but as expected the style transfer is completely dependent on the style image used. CycleGAN does look like it is able to capture the general style of the learned art movements, e.g. baroque and renaissance are dark, and impressionism is colorful. StarGAN unfortunately experiences modal collapse. In the examples, either the images become complete random splatter, or it is not able to find a correct mapping between the content of different images, e.g. in one image the face is mapped to the back. Looking at the different epochs, it seems that after more epochs the stylization is stronger. All in all, the results are very disappointing as none of the images look like they're a painting from a different time.

Quantitative Evaluation

To evaluate the trained models, there exist several metrics to do this with, as discussed in sec. 2.2.5. Before applying the evaluation metrics there needs to be an adequate dataset to do meaningful measurements on. Two datasets are considered for this purpose:

1. Arbitrary Style Transfer Image Quality Assessment Database (AST-IQAD) is a set specifically made to measure style transfer. [108] It constructs the set around a several inter-subjective characteristics and categories. This means that these criteria of subjective evaluation are mostly agreed upon across a group of people. Among those are: color tone, brush stroke, distribution of objects, and contents. While it also declares a set of style images, those will not be used.

2. Since the content of the problem of this thesis only focuses around persons and the AST-IQAD dataset works with different kinds of content, a custom dataset is created that focuses around persons. This is created the same way the style transfer datasets were created. The query images and results are shown in Fig. 3.13

For the evaluation, AdaIN cycles through the style images that it was trained on to use as input style images. The perceptual distance needs a content and style image to be able to make an evaluation. For AdaIN, it is clear what needs to be used here, but for the other models this metric seems useless. However, the dataset that CycleGAN and StarGAN were trained can be

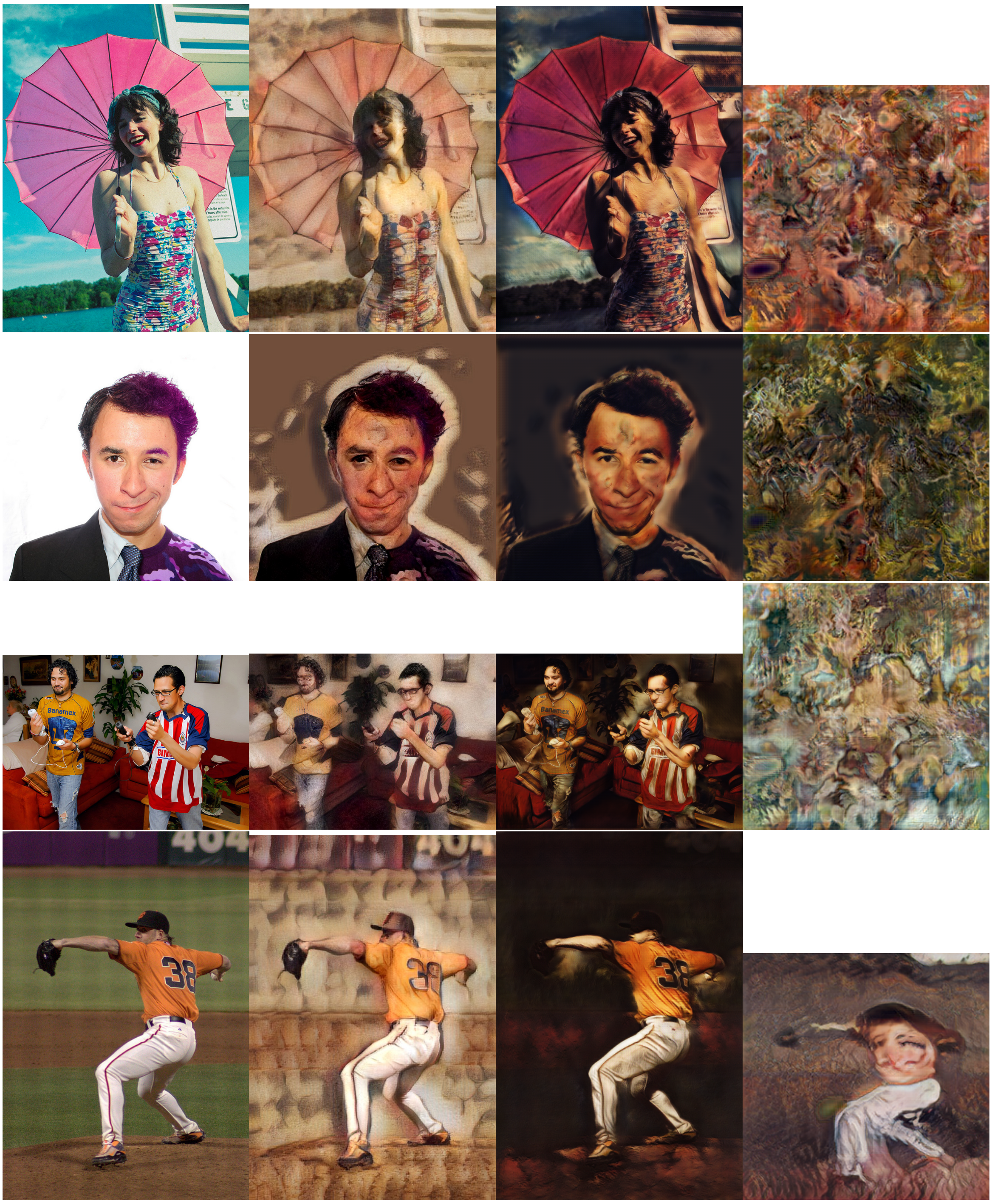


Figure 3.12: AdaIN abstracts the features more than CycleGAN, while StarGAN experiences modal collapse. Left is the content image. The middle-left is AdaIN using a renaissance style image. The middle-right is CycleGAN using the baroque style. The right is StarGAN impressionism.

(a) Query images

(b) Resulting dataset

Figure 3.13: The custom dataset used for style transfer evaluation consisting of 200 images.

used as style image for this. The style features of the generated images should still be similar as the ones it was trained on. These same datasets are also used for the real image distribution needed for FID and LPIPS.

In Table 3.2, the results of the evaluation are available. One model does not clearly seem to outperform the others. In fact, a pattern arises where AdaIN does well with Perceptual Distance (PD), CycleGAN does well with FID, StarGAN does well with IS and LPIPS have similar results for all. Impressionism does the best out of all of the styles. Table 3.2 shows the same result, but grouped by model. This shows the dataset used to evaluate the models, does not seem to have an influence on the evaluation.

3.1.5 Discussion

While the images are clearly stylized to look vaguely like the style of an artwork, it cannot be said that they belong in the same domain as the art movements. The stylized images can still be useful to augment the COCO dataset as the question whether stylized images can increase the evaluation results is still a useful one to ask. It is obvious that the used evaluation metrics for style transfer are not very helpful. Theoretically, they make complete sense, they do not at all give a good reading on the quality of the images. The numbers vary greatly, but this variance cannot be seen in the qualitative evaluation. StarGAN, which experienced modal collapse, was still able to score high for IS. Ironically, StarGAN, while not retaining the content, does have the better oil painting characteristics. The identity image as show in fig. **??** looks like modern art. So, somewhere, the model does approach some kind of human like abstraction, or at least, as seen in abstract art. Perhaps, how artists make abstractions can be used as an inductive bias in future models.

Table 3.2: Performance comparison of Style Transfer measured by various metrics grouped by dataset; Perceptual Distance (PD), Inception score (IS), Fréchet Inception Distance (FID) and Learned Perceptual Image Patch Similarity (LPIPS).

Method	Baroque				Impressionism				Renaissance			
	PD	IS	FID	LPIPS	PD	IS	FID	LPIPS	PD	IS	FID	LPIPS
AST-IQAD Dataset												
AdaIN	10.734	8.975	7.95E+88	0.626	**10.671**	8.453	4.89E+91	0.710	**10.746****	6.717	1.11E+89	**0.696****
CycleGAN	**10.514****	10.850	**-9.11E+83**	0.633	18.128	10.046	**2.31E+89**	**0.721**	12.258	9.878	**-2.07E+80**	0.689
StarGAN	15.767	**1.272****	3.93E+98	**0.651**	19.821	**1.223***	3.47E+96	0.713	17.665	**1.452****	1.21E+99	0.680
Custom Dataset												
AdaIN	**10.570**	6.639	2.72E+85	**0.654****	**10.122***	4.974	-8.70E+86	**0.737***	**11.472**	5.156	**2.89E+77****	**0.693**
CycleGAN	14.316	7.137	**-1.52E+83****	0.635	14.263	6.047	**-3.95E+64***	0.711	12.933	7.825	1.06E+87	0.678
StarGAN	19.178	**1.316**	-2.40E+95	0.648	17.728	**1.350**	1.85E+85	0.709	19.380	**1.456**	-4.17E+94	0.680

* the best result overall.

** the best result for the style.

Table 3.3: Performance comparison of Style Transfer measured by various metrics grouped by model; Perceptual Distance (PD), Inception score (IS), Fréchet Inception Distance (FID) and Learned Perceptual Image Patch Similarity (LPIPS).

Method	Baroque				Impressionism				Renaissance			
	PD	IS	FID	LPIPS	PD	IS	FID	LPIPS	PD	IS	FID	LPIPS
AdaIN												
AST-IQAD Dataset	10.734	8.975	7.95E+88	0.626	10.671	8.453	4.89E+91	0.710	**10.746**	6.717	1.11E+89	**0.696**
Custom Dataset	**10.570**	**6.639**	**2.72E+85**	**0.654**	**10.122**	**4.974**	**-8.70E+86**	**0.737**	11.472	**5.156**	**2.89E+77**	0.693
CycleGAN												
AST-IQAD Dataset	**10.514**	10.850	-9.11E+83	0.633	18.128	10.046	2.31E+89	**0.721**	**12.258**	9.878	**-2.07E+80**	**0.689**
Custom Dataset	14.316	**7.137**	**-1.52E+83**	**0.635**	**14.263**	**6.047**	**-3.95E+64**	0.711	12.933	**7.825**	1.06E+87	0.678
StarGAN												
AST-IQAD Dataset	**15.767**	**1.272**	3.93E+98	**0.651**	19.821	**1.223**	3.47E+96	**0.713**	**17.665**	**1.452**	1.21E+99	**0.680**
Custom Dataset	19.178	1.316	**-2.40E+95**	0.648	**17.728**	1.350	**1.85E+85**	0.709	19.380	1.456	**-4.17E+94**	**0.680**

Figure 3.14: The AFHQ dataset consists of images that are close-ups of animals.

The question remains why the style transfer algorithms aren't able to make correct mappings between different styles. A first observation was discussed in sec. 3.1.2. It's a mistake to consider an art movement as a style, as even within the different art movements and realistic photographs there's a big variation in styles. There can be different lighting, different brush stroke, different camera filter, different lines and different form. There are plenty of things that can vary to make a distinct style. It should be considered whether some things categorized as content now should instead be considered part of the style, like clothes. Whether clothes should be considered content or style can depend on which domains the mapping is searched for. Clothes change dramatically between the different time periods and this is clearly visible when comparing the artworks with photographs. In this context, they should be considered a style. While, when mapping within the same time period, they can be considered content. The same argument can be made for architecture.

When looking at the datasets that CycleGAN and StarGAN are trained on, it becomes obvious that the only success is made when the domain is extremely specific. As seen in Fig. 3.14, all the images contain the subject in the center of the image without any other content. The custom datasets for the training contain a much higher disparity. Perhaps it would be useful to transform different patches where the content is very similar with high certainty at a time, and then combine those to create the transformed image. This can potentially be done by training on a dataset of 3d models where a shader is applied to simulate a different art style. Instead of having to manually label thousands of images, it is possible to have several 3d models act out different poses and render them with different shaders. A network can be trained then to recognize when patches have similar content and apply the style when they do. This will mean that when using an arbitrary style, it might not always find a high similarity and the style transfer will not benefit from this.

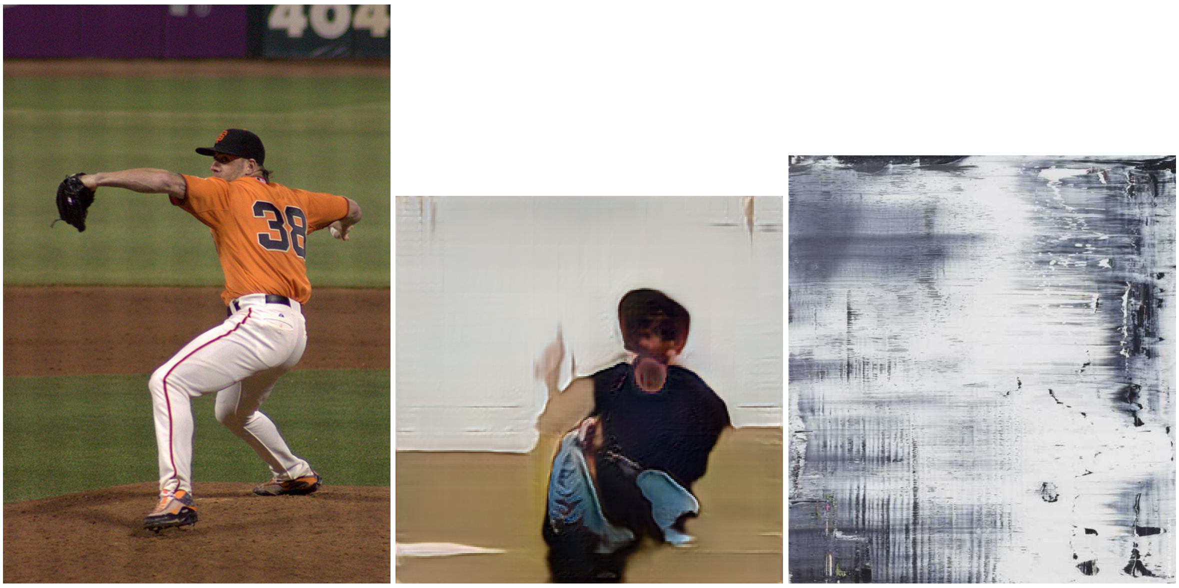
3.2 Baseline Pose Estimation

3.2.1 Choice of Model

Here again, quality is the most important criteria for performance. The current state-or-the-art is ViTPose [109]. The model is based on vision transformers. This makes it an obvious first choice. An overwhelming amount of models both in top-down as well as bottom-up architectures use HRNet [8] with the only difference being in pre-processing [110, 111] or post-processing [47, 112]. Since VitPose is a top-down architecture and to keep a variety of architectures, a bottom-up version of HRNet is selected. The best model in this family is SWAHR according to Chen et al. [25]. Other architectures were looked at, like KAPAO [113], which uses a single-stage architecture, but these were not performant enough to be considered.

3.2.2 Training

For the sake of learning the different algorithms, the training scrips were reverse engineered So, it was deemed appropriate to train the chosen models from scratch, so there's a plain network trained with the new setup for comparison. All training



(a) Input Image (b) Identity Image (c) Richter, Gerard, *Weiß*

Figure 3.15: An example of an image created by StarGAN that has oil painting qualities. A painting from Gerard Richter as comparison is shown.

was done using the default parameters.

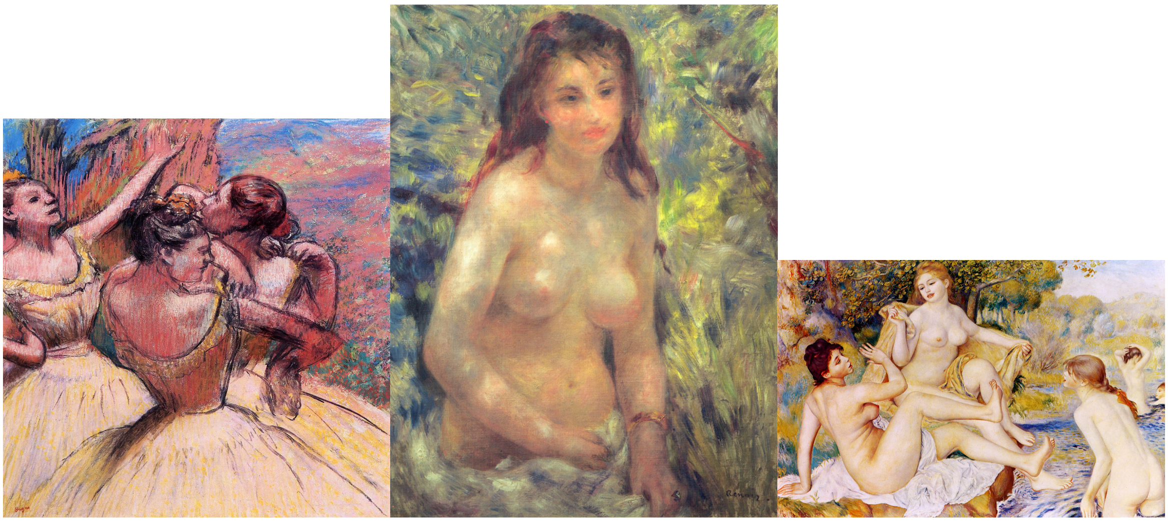
3.3 Pose Estimation after Applying Style Transfer to the COCO Dataset

Due to time constraints, the evaluation is only done on a subset of the COCO dataset. This set was created by randomly sampling 1000 images from the COCO dataset. The first baseline will establish how well the pre-trained models perform on a stylized COCO dataset. The tested pose estimators will be SWAHR and ViTPose, and each will use the trained weights mentioned in section 3.2.2. Since ViTPose is a top-down architecture, it will use the ground truth bounding box to extract the persons. They will both be tested on a styled version of the COCO dataset by CycleGAN and AdaIN. CycleGAN wil be applied for the 3 styles it was trained on; baroque, impressionism and renaissance. AdaIN uses the pre-trained model and uses 3 images of each of the previous styles to use as style image. The images were selected to best represent the style while also varying the content as shown in Fig. **??**. Each model uses the default parameters and at no time was the input image resized or otherwise distorted. This comes to a total of 24 combinations that will be tested.

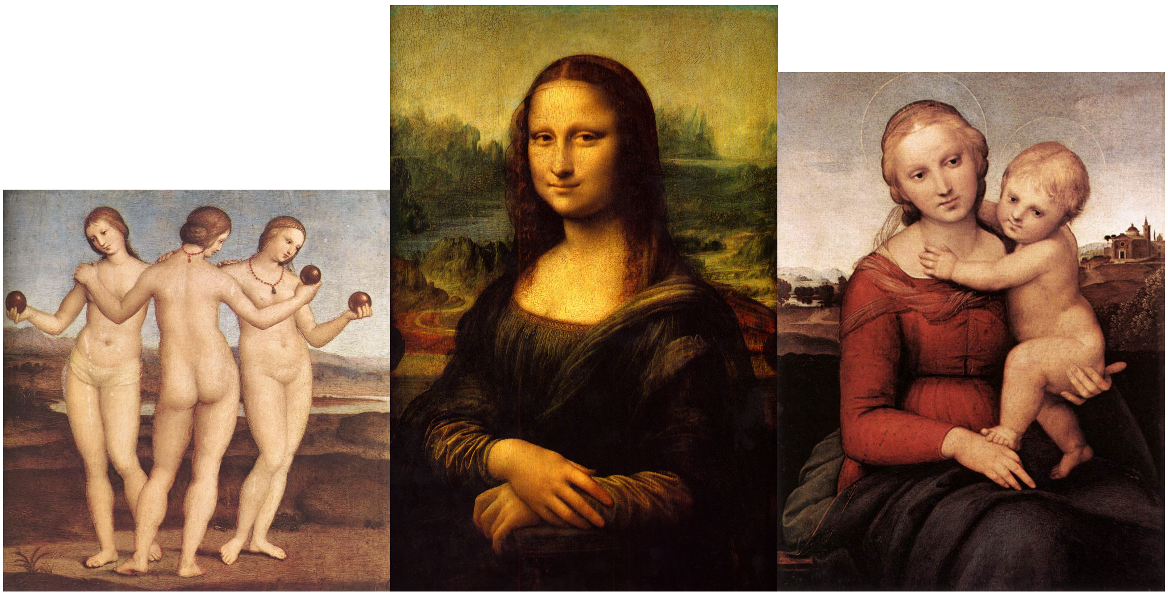
3.3.1 Results

From the available metrics the only ones that were useful for these measurements were the Average Precision/Recall. They're implemented as part of the COCO dataset and work with any dataset that's compatible with the COCO format. Of the other metrics, Percentage of Correct Parts (PCP) is unusable because it only applies to networks that detect the limbs as boxes instead of keypoints. The chosen pose estimation networks only work with keypoints. Percentage of Correct Keypoints (PCk) looked like it could be useable. However, PCKh needs a head bounding box, which is only available for the MPII dataset. While

(a) Baroque style images



(b) Impressionism style images



(c) Renaissance style images

Figure 3.16: The style images used for AdaIN during evaluation.

Table 3.4: Establishing a baseline for Pose Estimation on Artworks; measuring Average Precision/Recall (AP/AR). The COCO dataset is transformed with various Style Transfer models on which performance is measured from pre-trained pose-estimation models.

Method	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}	AR	AR^{50}	AR^{75}	AR^{M}	AR^{L}
					AdaIN					
SWAHR	0	0	0	0	0	0	0	0	0	0
ViTPose	0.026	0.057	0.020	0.017	0.041	0.340	0.568	0.337	0.187	0.539
					CycleGAN					
SWAHR	0	0	0	0	0	0	0	0	0	0
ViTPose	0.081	0.128	0.086	0.120	0.068	0.627	0.850	0.682	0.557	0.718

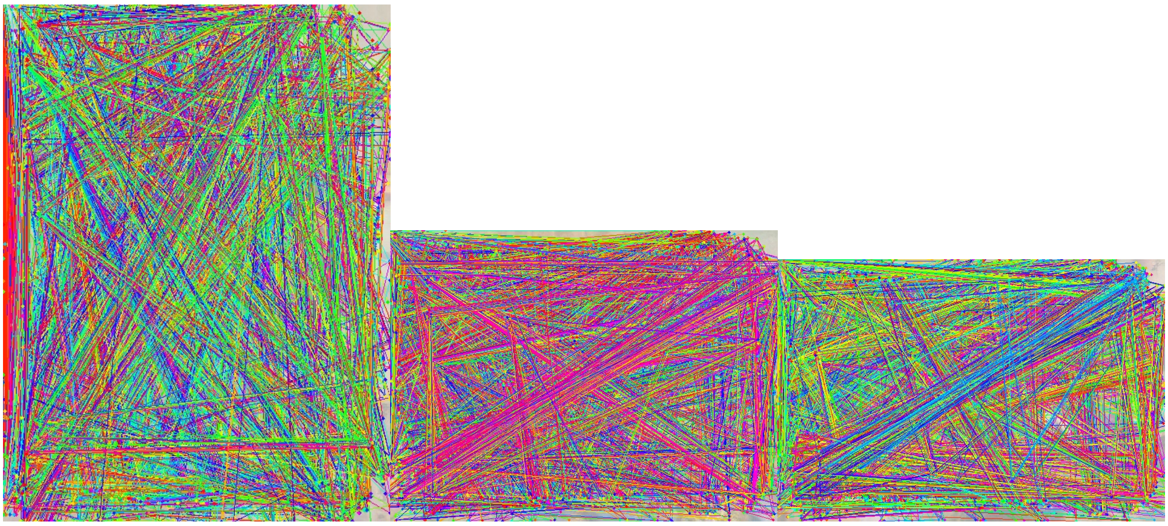
all the implementations only work with top-down architectures. They each asserts that the length of predicted persons should be the same as that of the ground truth. In a bottom-up architecture, it is possible to find more or less persons. The results shown in Table 3.5 are the average of different evaluations, and it becomes immediately evident that this method is not going to work. As seen in Fig. **??**, SWAHR is completely lost and can't find any good keypoints while ViTPose, having a high recall, still found some of the poses.

3.4 Pose Estimation on the Human-Art Dataset

As a second baseline, the Human-Art dataset contains a subset of annotated oil paintings compatible with the COCO format. The evaluation dataset contains 250 images with 900 annotated persons. This will give a insight in the performance of the pose estimation models on artworks. SWAHR and ViTPose will be validated, and the trained weights mentioned in section 3.2.2 as well as the pre-trained weights from the original papers will be used. The input image will not be resized or otherwise distorted, and the default parameters used. To confirm the premise that the pose estimation models perform less well on artworks, they are also validated on the COCO dataset. Because the other tests are only done on a subset of the COCO dataset, the models are also validated on this subset. This is a total of 8 combinations that will be validated.

3.4.1 Results

As mentioned in section 3.3.1, only the Average Precision/Recall will be measured. The table 3.5 shows the results of the measurements. It clearly shows that the models have inferior results on artworks than photographs by up to 20%. It also shows a significant difference between the pre-trained and self-trained models. However, for SWAHR the pre-trained model performed better by 6%, but for ViTPose, the self-trained model performs better by 2%. This difference well justifies the training of the models on the plain COCO dataset instead of using the pre-trained models to compare to. Thus going forward, the metrics will be compared to the self-trained models. This will give a more accurate picture of the improvements made. Notable as well is that despite ViTPose being the state-of-the-art, it performs worse than SWAHR on both datasets.



(a) SWAHR

(b) ViTPose

Figure 3.17: Examples of the keypoints found by the Pose Estimation networks.

Table 3.5: Establishing a baseline for Pose Estimation on Artworks; Average Precision/Recall (AP/AR). The table shows the performance of the pre-trained models on The COCO dataset measured and the Human-Art dataset.

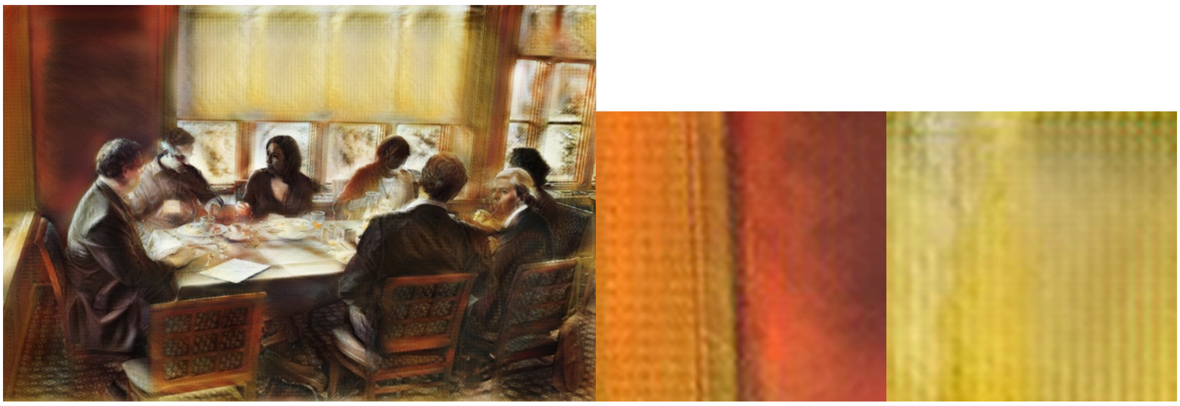
Method	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}	AR	AR^{50}	AR^{75}	AR^{M}	AR^{L}
COCO dataset										
Pre-trained SWAHR	0.687	0.881	0.748	0.639	0.757	0.737	0.904	0.788	0.670	0.828
SWAHR	0.620	0.830	0.684	0.604	0.653	0.710	0.891	0.765	0.640	0.803
Pre-trained ViTPose	0.588	0.832	0.641	0.573	0.629	0.723	0.906	0.782	0.682	0.7863
ViTPose	0.609	0.847	0.680	0.597	0.644	0.740	0.918	0.810	0.703	0.795
Human-Art Dataset										
Pre-trained SWAHR	0.528	0.759	0.565	0.099	0.573	0.593	0.635	0.629	0.177	0.635
SWAHR	0.492	0.742	0.536	0.058	0.539	0.563	0.784	0.606	0.109	0.605
Pre-trained ViTPose	0.380	0.656	0.385	0.108	0.420	0.571	0.803	0.620	0.279	0.599
ViTPose	0.406	0.682	0.415	0.130	0.445	0.591	0.818	0.632	0.306	0.619
Difference										
Pre-trained SWAHR	-0.159	-0.122	-0.183	-0.540	-0.184	-0.144	-0.269	-0.159	-0.493	-0.193
SWAHR	-0.128	-0.088	-0.148	-0.546	-0.114	-0.147	-0.107	-0.159	-0.531	-0.198
Pre-trained ViTPose	-0.208	-0.176	-0.256	-0.465	-0.209	-0.152	-0.103	-0.162	-0.403	-0.187
ViTPose	-0.203	-0.165	-0.265	-0.467	-0.199	-0.149	-0.100	-0.178	-0.397	-0.176

3.5 Discussion

The results show that the use of style transfer on the input image will not yield any good results. The models don't perform well on the stylized images, likely because they don't produce high-fidelity transformations. Putting a second algorithm in the pipeline creates an extra chance for error. However, ViTPose was still able to discern most of the poses; having a high recall, but seems to have hallucinated others; giving a low precision. This begs the question: what about this network makes it perform better than SWAHR here? Perhaps, it is merely able to deal with the artifacts left by the style transfer better, while SWAHR is completely confused by it? Fig. **??** shows these artifacts. Or, perhaps, it is merely because as a top-down algorithm, it has an unfair advantage in that it used the ground-truth bounding boxes to crop the image. (Todo: do small experiments with ViTPose and use entire image as bounding box)

The baseline on the plain COCO dataset confirms once more that the pose estimation models are inferior on artworks than photographs. It goes up as high as 50% for the medium areas, which makes sense as the smaller part of an image will also be more abstract as brush strokes become more prominent.

(a) AdaIN with impressionist middle image seen in fig. **??** as style image



(b) CycleGAN using impressionism as style

Figure 3.18: Examples of the keypoints found by the Pose Estimation networks.

4

Improving Pose Estimation with Style Transfer

Having established a baseline, it is now possible to search for improvements. In this chapter, 2 techniques will be explored to see if they can improve HPE. Using the same algorithms as seen in the previous chapter, they will now be used to: (1) transform an input artistic image to a photographic image to estimate poses on or (2) be trained with a dataset that is augmented with images that are transformed to different styles.

4.1 Pose Estimation after Style Transform

One option to predict poses on an artwork is to transform it first to photographic realism and let the plain model run on it. As previously seen in section **??**, the results on photographs are dramatically better. If artworks are successfully transformed to that style, there is no need to train a new model and the extensive amount of datasets created for this task are available. To validate this, SWAHR and ViTPose are run on the Human-Art dataset after it was transformed using AdaIN and CycleGAN. For CycleGAN, 3 styles are used to perform this task, namely baroque, impressionism and renaissance. AdaIN uses 3 style images for each style previously mentioned. Fig. **??** shows the selection made for this. Before transforming the artwork, the size is checked and resized to 1024 if either of the sides is bigger than that, and only then. This is done because some artworks in the dataset are quite large and cause Out-Of-Memory errors. Otherwise, no other distortions are applied. This adds the total number of tests to be up to 24.

4.1.1 Results

As seen in section **??**, here as well, style transfer is only more detrimental to the results. As seen in Table 4.1, the same observations can be made: Only ViTPose scores, but with very low precision and high recall.

4.2 Augmenting COCO Dataset for Pose Estimation Training

The second option that's been explored is the augmentation of the dataset with styled images. The chosen pose estimation algorithms, SWAHR and ViTPose, will be trained on several different stylized datasets. A combination of the COCO dataset and the stylized dataset is used, and one with only the stylized dataset. The stylized datasets are created by applying both CycleGAN and AdaIN to the COCO dataset. One with a mixture of the baroque, impressionism and renaissance models, and one with only the impressionism model. This results in a combination of 18 models. The experiments will be conducted on

Table 4.1: Performance of plain Pose Estimation models after Artwork is transformed with different Style Transfer models.

Method	**AP**	**AP**^{50}	**AP**^{75}	**AP**^{M}	**AP**^{L}	**AR**	**AR**^{50}	**AR**^{75}	**AR**^{M}	**AR**^{L}
AdaIN										
Trained on Baroque dataset										
SWAHR	0	0	0	0	0	0	0	0	0	
ViTPose	0.056	0.109	0.052	0.002	0.064	0.463	0.700	0.486	0.058	0.501
Trained on Impressionism dataset										
SWAHR	0	0	0	0	0	0	0	0	0	0
ViTPose	0.044	0.089	0.043	0.002	0.051	0.406	0.648	0.427	0.051	0.439
Trained on Renaissance dataset										
SWAHR	0	0	0	0	0	0	0	0	0	0
ViTPose	0.052	0.100	0.047	0.001	0.058	0.441	0.0679	0.457	0.045	0.477
CycleGAN										
Trained on Baroque dataset										
SWAHR	0	0	0	0	0	0	0	0	0	0
ViTPose	0.068	0.128	0.066	0.014	0.075	0.520	0.768	0.555	0.195	0.551
Trained on Impressionism dataset										
SWAHR	0	0	0	0	0	0	0	0	0	0
ViTPose	0.059	0.113	0.055	0.020	0.064	0.470	0.717	0.488	0.227	0.493
Trained on Renaissance dataset										
SWAHR	0	0	0	0	0	0	0	0	0	0
ViTPose	0.057	0.107	0.052	0.012	0.062	0.458	0.694	0.471	0.183	0.485

the COCO-dataset as well as the Human-Art dataset. While the problem specifically tries to improve the performance on artworks, it's still interesting to also validate the results on the COCO-dataset.

4.2.1 Creation of datasets

For each augmented dataset, the coco annotations file was used as template. All metadata of the file was kept while only changing the id and file name. To each id a number of several hundred billions was added depending on the style transfer model. The file name points to the new location of the stylized image. A stylized version of COCO was created from each style transfer model that CycleGAN was trained for discussed in section **??**, except impressionism for 2000 epochs. Other versions were created for AdaIN. Since AdaIN requires a style image, the images used for training the CycleGAN models were used for this purpose. The style dataset was cycled through to transform the COCO dataset with AdaIN. The decision to not use one image as a representation for each style was made so that the dataset is more generalized. Afterwards, a new annotation file was created from a mixture of baroque, impressionism and renaissance stylized images, and one of only the impressionism style. For each, a version was made which is appended to the COCO dataset and one that stand on its own. During training it was noticed that the stylized images were inverted, resulting in 2 models being trained on the inverted dataset. These were the COCO + mixed and mixed models.

4.2.2 Training

All models are trained with the default parameters initiating the weights with the pre-trained model. They're trained for 200 epochs and the models are saved from 100th epoch every 20 epochs. As a control, 2 models are trained from nought, which are the inverted mixed model and the COCO + impressionism model. These were trained for the default 300 epochs and were also saved from 100th epoch every 20 epochs.

4.2.3 Results

For the SWAHR network, shown in table 4.2, the best results are found with the model trained on the COCO + AdaIN mixed style transfer dataset. The second best network was trained on the COCO + CycleGAN mixed style transfer dataset. For the ViTPose network, the best results are with COCO + CycleGAN mixed and COCO + CycleGAN impressionism being the second best. For AdaIN, there's a falloff of 7 to 10% AP between the datasets with COCO and the ones without. For CycleGAN, this falloff is less; between 0.2 and 4% AP. The best precision is found using the SWAHR model, while ViTPose has the honor of having the best recall. Table 4.2 compares the best models with the baseline. It shows that the pre-trained SWAHR model has the best precision of all of the models and trained on the COCO + AdaIN Mixed style transfer dataset, SWAHR also has the second best precision. ViTPose trained on COCO + CycleGAN mixed style transfer dataset has the best recall. The networks trained from the ground up don't have any significant difference between the other networks.

The results on the Human-Art dataset are shown in table 4.4. Here, one dataset takes the crown. Both SWAHR as well as ViTPose have the best results for the models trained on the COCO + CycleGAN mixed style transfer dataset. The second best model for SWAHR is trained on the COCO + AdaIN mixed dataset and for ViTPose this is the one trained on COCO + CycleGAN impressionism. The falloff between the COCO and non-COCO datasets is between 5 to 9% AP for AdaIN, and 2 to 3% AP for CycleGAN. The best precision and recall belongs to the SWAHR models. Comparing the best models to the baseline, they still remain the best models overall with an increase of 3 to 5% AP. There's no significant difference between the non-initialized and bootstrapped networks.

4.3 Discussion

It is overwhelmingly obvious that trying to use style transfer to transform images to try to use pre-trained networks on is a catastrophic failure. Style transfer, or at least the models used in this thesis, does not have the capacities to convincingly transform a photograph to an artwork. It's difficult to believe that any of the styled images can be confused with an artwork by any reasonable person. There are several studies that confirm this: Chen et al. [18] and Wang et al. [19] calculate a deception score, which measure the believability of the fake images against the real images. Images from a set of stylized images and real artworks are shown to participants who need to determine whether it is real or fake. Table 4.7 shows that older networks have extremely bad performance on this with a meager 40% at best. While the newer models show a considerable improvement, they're still 20% below the real images. Other models, like Huang et al. [114] and Zhang et al. [115] ask participant to select the fake(s) from a group of images. They find that their models were able to confuse participants; participants were not able to distinct between fake and real images, while for older models this was not achieved. This confirms the observation that the used models are inadequate, but gives hopeful results for future research with state-of-the-art style transfer models.

Table 4.2: Performance of different Pose Estimation models trained on Style Transformed datasets on COCO dataset.

Method	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}	AR	AR^{50}	AR^{75}	AR^{M}	AR^{L}
AdaIN										
Trained on COCO + Mixed Style Transfer										
SWAHR	**0.679***	**0.874***	0.735	**0.628***	0.751	**0.732**	**0.902**	0.782	0.651	0.824
ViTPose	0.618	0.859	0.685	0.599	0.661	0.748	0.924	0.816	0.709	0.805
Trained on COCO + Impressionism Style Transfer										
SWAHR	0.669	0.862	0.733	0.607	**0.755***	0.729	**0.902**	0.782	0.651	**0.834**
ViTPose	0.609	0.843	0.664	0.590	0.654	0.742	0.916	0.801	0.702	0.799
Trained on Mixed Style Transfer										
SWAHR	0.603	0.843	0.661	0.535	0.704	0.676	0.882	0.726	0.586	0.794
ViTPose	0.518	0.783	0.557	0.492	0.573	0.669	0.880	0.726	0.617	0.739
Trained on Impressionism Style Transfer										
SWAHR	0.591	0.830	0.654	0.527	0.688	0.663	0.873	0.716	0.574	0.780
ViTPose	0.497	0.784	0.531	0.463	0.564	0.650	0.874	0.710	0.594	0.728
CycleGAN										
Trained on COCO + Mixed Style Transfer										
SWAHR	0.672	0.863	**0.737***	0.618	0.747	**0.732**	**0.902**	**0.787**	**0.660**	0.827
ViTPose	**0.635**	**0.861**	0.697	0.616	**0.681**	**0.763***	**0.925***	**0.825***	0.723	**0.820**
Trained on COCO + Impressionism Style Transfer										
SWAHR	0.663	0.862	0.724	0.606	0.743	0.714	0.889	0.764	0.637	0.815
ViTPose	0.633	0.859	**0.701**	**0.618**	0.670	0.761	0.922	0.828	**0.725***	0.812
Trained on Mixed Style Transfer										
SWAHR	0.653	0.858	0.711	0.609	0.714	0.716	0.898	0.764	0.647	0.807
ViTPose	0.595	0.844	0.654	0.586	0.628	0.731	0.912	0.795	0.698	0.780
Trained on Impressionism Style Transfer										
SWAHR	0.661	0.864	0.717	0.621	0.719	0.718	0.896	0.765	0.656	0.802
ViTPose	0.591	0.841	0.643	0.582	0.619	0.727	0.910	0.790	0.695	0.773

* the best result overall.

Table 4.3: Comparing the best models from 4.2 with the baseline metrics found in table 3.5.

Method	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}	AR	AR^{50}	AR^{75}	AR^{M}	AR^{L}
Pre-trained SWAHR	**0.687***	**0.881***	**0.748***	**0.639***	**0.757***	0.737	0.904	0.788	0.670	**0.828***
Pre-trained ViTPose	0.588	0.832	0.641	0.573	0.629	0.723	0.906	0.782	0.682	0.7863
SWAHR	0.620	0.830	0.684	0.604	0.653	0.710	0.891	0.765	0.640	0.803
ViTPose	0.609	0.847	0.680	0.597	0.644	0.740	0.918	0.810	0.703	0.795
SWAHR COCO + AdaIN Mixed	**0.679****	**0.874****	**0.735****	**0.628****	**0.751****	0.732	0.902	0.782	0.651	**0.824****
ViTPose COCO + CycleGAN Mixed	0.635	0.861	0.697	0.616	0.681	**0.763***	**0.925***	**0.825***	**0.723***	0.820

* the best result overall.

** the best result without pre-trained models.

Table 4.4: Performance of different Pose Estimation models trained on Style Transferred datasets on Human-Art dataset.

Method	**AP**	**AP**^{50}	**AP**^{75}	**AP**^{M}	**AP**^{L}	**AR**	**AR**^{50}	**AR**^{75}	**AR**^{M}	**AR**^{L}
AdaIN										
Trained on COCO + Mixed Style Transfer										
SWAHR	0.549	**0.791***	0.600	0.065	**0.602***	0.622	0.834	0.668	0.141	0.667
ViTPose	0.420	0.724	0.440	**0.151***	0.460	0.600	0.843	0.650	0.300	0.630
Trained on COCO + Impressionism Style Transfer										
SWAHR	0.540	0.779	0.576	0.071	0.591	0.612	0.822	0.646	0.156	0.655
ViTPose	0.421	0.706	0.430	0.149	0.458	0.600	0.831	0.641	0.303	0.629
Trained on Mixed Style Transfer										
SWAHR	0.492	0.750	0.525	0.048	0.547	0.581	0.811	0.625	0.142	0.622
ViTPose	0.332	0.627	0.316	0.079	0.372	0.522	0.784	0.559	0.223	0.551
Trained on Impressionism Style Transfer										
SWAHR	0.488	0.738	0.524	0.058	0.543	0.581	0.804	0.624	0.153	0.621
ViTPose	0.321	0.600	0.302	0.094	0.355	0.514	0.765	0.539	0.232	0.542
CycleGAN										
Trained on COCO + Mixed Style Transfer										
SWAHR	**0.553***	0.789	**0.604***	**0.122**	0.598	**0.629***	0.839	**0.677***	**0.208**	**0.669***
ViTPose	**0.439**	**0.726**	**0.458**	0.140	**0.481**	0.617	0.844	0.661	0.324	**0.646**
Trained on COCO + Impressionism Style Transfer										
SWAHR	0.522	0.778	0.556	0.113	0.565	0.590	0.819	0.628	0.173	0.630
ViTPose	0.438	0.724	0.448	0.147	0.479	**0.619**	**0.846***	**0.664**	**0.358***	0.645
Trained on Mixed Style Transfer										
SWAHR	0.524	0.779	0.559	0.102	0.569	0.613	**0.843**	0.645	0.200	0.652
ViTPose	0.405	0.696	0.419	0.148	0.442	0.590	0.829	0.639	0.338	0.615
Trained on Impressionism Style Transfer										
SWAHR	0.505	0.761	0.539	0.116	0.546	0.587	0.822	0.622	**0.208**	0.623
ViTPose	0.407	0.694	0.412	**0.151***	0.444	0.590	0.828	0.631	0.341	0.615

* the best result overall.

Table 4.5: Comparing the best models from 4.4 with the baseline metrics found in table 3.5.

Method	**AP**	**AP**^{50}	**AP**^{75}	**AP**^{M}	**AP**^{L}	**AR**	**AR**^{50}	**AR**^{75}	**AR**^{M}	**AR**^{L}
Pre-trained SWAHR	0.528	0.759	0.565	0.099	0.573	0.593	0.635	0.629	0.177	0.635
Pre-trained ViTPose	0.380	0.656	0.385	0.108	0.420	0.571	0.803	0.620	0.279	0.599
SWAHR	0.492	0.742	0.536	0.058	0.539	0.563	0.784	0.606	0.109	0.605
ViTPose	0.406	0.682	0.415	0.130	0.445	0.591	0.818	0.632	0.306	0.619
SWAHR COCO + CycleGAN Mixed	**0.553***	**0.789***	**0.604***	0.122	**0.598***	**0.629***	0.839	**0.677***	0.208	**0.669***
ViTPose COCO + CycleGAN Mixed	0.439	0.726	0.458	**0.140***	0.481	0.617	**0.844***	0.661	**0.324***	0.646

* the best result overall.

** the best result without pre-trained models.

Table 4.6: The deception score of different models calculated by Chen et al. [18] and Wang et al. [19].

Paper	**WikiArt**	**Theirs**	**AdaIN**	**WCT [117]**	**LST [118]**	**SANet [119]**
Chen et al.	0.875	0.624	0.363	0.099	0.125	0.161
Wang et al.	0.784	0.568	0.241	0.172	0.408	0.346

Table 4.7: The deception score of different models calculated by Chen et al. [18] and Wang et al. [19].

Paper	**WikiArt**	**Theirs**	**AdaIN**	**WCT [117]**	**LST [118]**	**SANet [119]**
Chen et al.	0.875	0.624	0.363	0.099	0.125	0.161
Wang et al.	0.784	0.568	0.241	0.172	0.408	0.346

During training, a plain model was trained as a control for the fidelity of the reverse-engineered implementation. On the COCO-dataset, ViTPose improved on both the control and pre-trained models. The results for CycleGAN don't seem to be an improvement when comparing with the pre-trained model, but there are improvements compared to the control. This could mean that if the difference in training can be pin-pointed, the performance on the COCO-dataset for pre-trained models could potentially be increased. However, this is not a guarantee. On the other hand, the performances on the Human-Art dataset have increased compared to both baselines for both architectures. The models with the best results are those that combine the COCO dataset with a mixture of different styles and SWAHR shows the best performance of the pose estimation algorithms.

(Todo: check results of both these models on the entire COCO dataset and entire Human-Art dataset)

During the training, for every 20 epochs the networks were evaluated after 100 epochs. Table **??** show that after 100 iterations for both SWAHR and ViTPose the network only marginally increased; around 2%. While this is great for fine-tuning the network, for comparing architectures this does not seem necessary. The same conclusions would have been drawn when keeping to only 100 epochs. In Table **??**, the metrics are shown after 100 epochs to illustrate this point for CycleGAN, as well as Fig. **??** for ViTPose.

Despite being state-of-the-art, ViTPose has a lower performance than SWAHR here. The evaluation is only on a subset of the COCO-dataset, which might explain it. According to Dosovitskiy et al., [116] Vision Transformers do not benefit from the inductive biases inherent to CNNs. To make up for that they need to be trained on a bigger dataset. With the augmentation of the COCO-dataset, the training size was doubled. The increased performance might just only be because of a larger dataset.

According to several surveys, bottom-up architectures are less precise then top-down architectures. Is this because top-down architectures are trained to only find one pose per ground truth in the found bounding box while bottom-up algorithms can find more poses per ground truth? This can skew the precision as there are now more false negatives. The cropping of the image also removes a lot of information that could potentially be relevant, like sitting on a horse or perspective. These could be clues that can help the algorithm more accurately do predictions, but how well can a network train for this? Perhaps 2d is limited in that sense.

4.4 Related Papers

Enhancing Human Pose Estimation in Ancient Vase Paintings via Perceptually-grounded Style Transfer Learning [102]

4.4.1 Results

Compare results with related paper

5

Evaluation in the Wild

This chapter will run the algorithms on the Art Collection from RMFAB as well as some that didn't qualify, but of which the results on a small dataset is still interesting. From the Art Collection a set of images is chosen that have the highest rate of failure. These include images with overlapping persons, occlusion, deformation, ...

5.1 RMFAB Dataset

What choices were made to establish the RMFAB dataset

5.2 Tests

Explanation of what tests were run UGATIT was adapted to randomize the B image in the dataset. We use the unaligned dataset from CycleGAN to do this

5.3 Results

What are the results from the tests

5.4 Discussion

Is it even possible to encode the information in an image correctly. When you look at several painting from monet where he draws the same "content" at different times but in the same season there's still a significant difference between them. It could be that the mood of the artist changed that caused him to choose a different color, or that some lighting or other influences outside the frame change its "style". Like with Claude Monet, who has many different paintings of the same subject.

Talk a bit about HD pictures and models

Discuss code:

Discuss the code and discussions during implementation. Also, what could be done differently (own implementation) Discuss how there are many different ways to "choose" the style (change model (cyclegan), choose number (stargan), use style image) This could be solved by creating a new interface for the styles with each their own options, etc Make everything highly configurable

Transformation interesting for future research.

Suppose you take a CNN: it will do convolutions, max pooling until you get as output a vector which you can use for cross-entropy, softmax loss. This has the entire image as perceptive field, with every layer the perceptive field grows bigger (check if this is true) until the last layer has the entire image in its field. Suppose that you want to know the coordinates of the object found, all you would need to know is what point in the neural network the perceptive field can see the object. From that point in the network it would be convenient to have the coordinates marked somewhere so that the object can be found at different scales. Meaning that for every layer it branches to a subnetwork or as another entrance for backpropogation (as with RNN). Is this how HRNet works? (research) Why is dataset thrice the size as the original dataset?

Discuss the flaws of mmpose: log_processor doesn't give enough info, eval code during runtime, all centered around configuration, but misses ease of programming. HAs train_loop, val_loop variables for extra confusion Don't have your code add prefixes to output dirs or anything else. It only causes confusion. It's also difficult to add new stuff, because the hooks don't provide enough information meaning hacks need to be implemented. When resuming a network, the iterations continue from previous session, but if the new session has a bigger or smaller world, those iterations don't match the new world size. How does multiple distribution sessions work?

On how to do research: would a method of research like gradient descent where one does a quick research paper of to check improvements and only proceeds in a certain direction when improvements are significant. Instead of researching every single variable.

Use a different backend and not visdom. It's difficult to alter test results with visdom, like removing unnecessary domains.

float16 for quicker warmup

When comparing ViTPose and SWAHR we should note that there is a HRNet version with vision transformers and maybe even do some tests.

Conclusions

References

[1] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, “2d human pose estimation: New benchmark and state of the art analysis,” in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3686–3693.

[2] Y. Chen, Y. Tian, and M. He, “Monocular human pose estimation: A survey of deep learning-based methods,” *CoRR*, vol. abs/2006.01423, 2020. [Online]. Available: https://arxiv.org/abs/2006.01423

[3] C. Zheng, W. Wu, T. Yang, S. Zhu, C. Chen, R. Liu, J. Shen, N. Kehtarnavaz, and M. Shah, “Deep learning-based human pose estimation: A survey,” *CoRR*, vol. abs/2012.13392, 2020. [Online]. Available: https://arxiv.org/abs/2012.13392

[4] A. Toshev and C. Szegedy, “DeepPose: Human pose estimation via deep neural networks,” in *2014 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, jun 2014. [Online]. Available: https://doi.org/10.1109%2Fcvpr.2014.214

[5] V. Ramakrishna, D. Munoz, M. Hebert, J. Andrew Bagnell, and Y. Sheikh, “Pose machines: Articulated pose estimation via inference machines,” in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 33–47.

[6] S. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, “Convolutional pose machines,” *CoRR*, vol. abs/1602.00134, 2016. [Online]. Available: http://arxiv.org/abs/1602.00134

[7] A. Newell, K. Yang, and J. Deng, “Stacked hourglass networks for human pose estimation,” *CoRR*, vol. abs/1603.06937, 2016. [Online]. Available: http://arxiv.org/abs/1603.06937

[8] K. Sun, B. Xiao, D. Liu, and J. Wang, “Deep high-resolution representation learning for human pose estimation,” *CoRR*, vol. abs/1902.09212, 2019. [Online]. Available: http://arxiv.org/abs/1902.09212

[9] Y. Chen, C. Shen, X. Wei, L. Liu, and J. Yang, “Adversarial posenet: A structure-aware convolutional network for human pose estimation,” *CoRR*, vol. abs/1705.00389, 2017. [Online]. Available: http://arxiv.org/abs/1705.00389

[10] L. A. Gatys, A. S. Ecker, and M. Bethge, “Image style transfer using convolutional neural networks,” *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2414–2423, 2016. [Online]. Available: https://api.semanticscholar.org/CorpusID:206593710

[11] D. Ulyanov, V. Lebedev, A. Vedaldi, and V. S. Lempitsky, “Texture networks: Feed-forward synthesis of textures and stylized images,” *CoRR*, vol. abs/1603.03417, 2016. [Online]. Available: http://arxiv.org/abs/1603.03417

[12] J. Johnson, A. Alahi, and L. Fei-Fei, “Perceptual losses for real-time style transfer and super-resolution,” *CoRR*, vol. abs/1603.08155, 2016. [Online]. Available: http://arxiv.org/abs/1603.08155

[13] D. Ulyanov, A. Vedaldi, and V. Lempitsky, “Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 4105–4113.

[14] Y. Taigman, A. Polyak, and L. Wolf, “Unsupervised cross-domain image generation,” *CoRR*, vol. abs/1611.02200, 2016. [Online]. Available: http://arxiv.org/abs/1611.02200

[15] M. Liu, T. M. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," *CoRR*, vol. abs/1703.00848, 2017. [Online]. Available: http://arxiv.org/abs/1703.00848

[16] W. Zhou, H. Li, and Q. Tian, "Recent advance in content-based image retrieval: A literature survey," *CoRR*, vol. abs/1706.06064, 2017. [Online]. Available: http://arxiv.org/abs/1706.06064

[17] B. Saleh and A. M. Elgammal, "Large-scale classification of fine-art paintings: Learning the right metric on the right feature," *CoRR*, vol. abs/1505.00855, 2015. [Online]. Available: http://arxiv.org/abs/1505.00855

[18] W. Chen, Y. Liu, W. Wang, E. M. Bakker, T. Georgiou, P. W. Fieguth, L. Liu, and M. S. Lew, "Deep image retrieval: A survey," *CoRR*, vol. abs/2101.11282, 2021. [Online]. Available: https://arxiv.org/abs/2101.11282

[19] Z. Wang, Z. Zhang, L. Zhao, Z. Zuo, A. Li, W. Xing, and D. Lu, "Aesust: Towards aesthetic-enhanced universal style transfer," 2022.

[20] M.-C. Marinescu, A. Reshetnikov, and J. M. López, "Improving object detection in paintings based on time contexts," in *2020 International Conference on Data Mining Workshops (ICDMW)*, 2020, pp. 926–932.

[21] M. Sabatelli, N. Banar, M. Cocriamont, E. Coudyzer, K. Lasaracina, W. Daelemans, P. Geurts, and M. Kestemont, "Advances in digital music iconography: Benchmarking the detection of musical instruments in unrestricted, non-photorealistic images from the artistic domain," *Digital Humanities Quarterly*, vol. 15, no. 1, February 2021.

[22] R. M. of Fine Arts Belgium. (2024) Opac fabritius. [Online]. Available: https://www.opac-fabritius.be/

[23] T. L. Munea, Y. Z. Jembre, H. T. Weldegebriel, L. Chen, C. Huang, and C. Yang, "The progress of human pose estimation: A survey and taxonomy of models applied in 2d human pose estimation," *IEEE Access*, vol. 8, pp. 133 330–133 348, 2020.

[24] W. Liu, Q. Bao, Y. Sun, and T. Mei, "Recent advances in monocular 2d and 3d human pose estimation: A deep learning perspective," *CoRR*, vol. abs/2104.11536, 2021. [Online]. Available: https://arxiv.org/abs/2104.11536

[25] H. Chen, R. Feng, S. Wu, H. Xu, F. Zhou, and Z. Liu, "2d human pose estimation: a survey," *Multimedia Systems*, pp. 1–24, 2022.

[26] A. Jain, J. Tompson, M. Andriluka, G. W. Taylor, and C. Bregler, "Learning human pose estimation features with convolutional networks," 2014.

[27] Z. Luo, Z. Wang, Y. Huang, T. Tan, and E. Zhou, "Rethinking the heatmap regression for bottom-up human pose estimation," *CoRR*, vol. abs/2012.15175, 2020. [Online]. Available: https://arxiv.org/abs/2012.15175

[28] S. Ju, M. Black, and Y. Yacoob, "Cardboard people: a parameterized model of articulated image motion," in *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, 1996, pp. 38–44.

[29] T. Cootes, C. Taylor, D. Cooper, and J. Graham, "Active shape models-their training and application," *Computer Vision and Image Understanding*, vol. 61, no. 1, pp. 38–59, 1995. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1077314285710041

[30] H. Sidenbladh, F. De la Torre, and M. Black, "A framework for modeling the appearance of 3d articulated figures," in *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580)*, 2000, pp. 368–375.

[31] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. Black, "Smpl: a skinned multi-person linear model," vol. 34, 11 2015.

[32] S. Johnson and M. Everingham, "Clustered pose and nonlinear appearance models for human pose estimation," in *British Machine Vision Conference*, 2010. [Online]. Available: https://api.semanticscholar.org/CorpusID:7318714

[33] ——, "Learning effective human pose estimation from inaccurate annotation," in *CVPR 2011*, 2011, pp. 1465–1472.

[34] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: common objects in context," *CoRR*, vol. abs/1405.0312, 2014. [Online]. Available: http://arxiv.org/abs/1405.0312

[35] B. Sapp and B. Taskar, "Modec: Multimodal decomposable models for human pose estimation," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3674–3681.

[36] J. Li, C. Wang, H. Zhu, Y. Mao, H. Fang, and C. Lu, "Crowdpose: Efficient crowded scenes pose estimation and A new benchmark," *CoRR*, vol. abs/1812.00324, 2018. [Online]. Available: http://arxiv.org/abs/1812.00324

[37] X. Ju, A. Zeng, J. Wang, Q. Xu, and L. Zhang, "Human-art: A versatile human-centric dataset bridging natural and artificial scenes," 2023.

[38] G. Pons-Moll and B. Rosenhahn, *Model-Based Pose Estimation*. London: Springer London, 2011, pp. 139–170. [Online]. Available: https://doi.org/10.1007/978-0-85729-997-0_9

[39] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., vol. 25. Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

[40] J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik, "Human pose estimation with iterative error feedback," *CoRR*, vol. abs/1507.06550, 2015. [Online]. Available: http://arxiv.org/abs/1507.06550

[41] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *CoRR*, vol. abs/1409.4842, 2014. [Online]. Available: http://arxiv.org/abs/1409.4842

[42] J. Tompson, A. Jain, Y. LeCun, and C. Bregler, "Joint training of a convolutional network and a graphical model for human pose estimation," *CoRR*, vol. abs/1406.2984, 2014. [Online]. Available: http://arxiv.org/abs/1406.2984

[43] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, "Efficient object localization using convolutional networks," *CoRR*, vol. abs/1411.4280, 2014. [Online]. Available: http://arxiv.org/abs/1411.4280

[44] W. Yang, S. Li, W. Ouyang, H. Li, and X. Wang, "Learning feature pyramids for human pose estimation," *CoRR*, vol. abs/1708.01101, 2017. [Online]. Available: http://arxiv.org/abs/1708.01101

[45] C. Chou, J. Chien, and H. Chen, "Self adversarial training for human pose estimation," *CoRR*, vol. abs/1707.02439, 2017. [Online]. Available: http://arxiv.org/abs/1707.02439

[46] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[47] B. Cheng, B. Xiao, J. Wang, H. Shi, T. S. Huang, and L. Zhang, "Bottom-up higher-resolution networks for multi-person pose estimation," *CoRR*, vol. abs/1908.10357, 2019. [Online]. Available: http://arxiv.org/abs/1908.10357

[48] C. Yu, B. Xiao, C. Gao, L. Yuan, L. Zhang, N. Sang, and J. Wang, "Lite-hrnet: A lightweight high-resolution network," *CoRR*, vol. abs/2104.06403, 2021. [Online]. Available: https://arxiv.org/abs/2104.06403

[49] Y. Yuan, R. Fu, L. Huang, W. Lin, C. Zhang, X. Chen, and J. Wang, "Hrformer: High-resolution transformer for dense prediction," *CoRR*, vol. abs/2110.09408, 2021. [Online]. Available: https://arxiv.org/abs/2110.09408

[50] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014.

[51] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *CoRR*, vol. abs/1411.1784, 2014. [Online]. Available: http://arxiv.org/abs/1411.1784

[52] C. Chou, J. Chien, and H. Chen, "Self adversarial training for human pose estimation," *CoRR*, vol. abs/1707.02439, 2017. [Online]. Available: http://arxiv.org/abs/1707.02439

[53] U. Iqbal and J. Gall, "Multi-person pose estimation with local joint-to-person associations," *CoRR*, vol. abs/1608.08526, 2016. [Online]. Available: http://arxiv.org/abs/1608.08526

[54] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *CoRR*, vol. abs/1506.01497, 2015. [Online]. Available: http://arxiv.org/abs/1506.01497

[55] L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. V. Gehler, and B. Schiele, "Deepcut: Joint subset partition and labeling for multi person pose estimation," *CoRR*, vol. abs/1511.06645, 2015. [Online]. Available: http://arxiv.org/abs/1511.06645

[56] H. Fang, S. Xie, and C. Lu, "RMPE: regional multi-person pose estimation," *CoRR*, vol. abs/1612.00137, 2016. [Online]. Available: http://arxiv.org/abs/1612.00137

[57] G. Papandreou, T. Zhu, N. Kanazawa, A. Toshev, J. Tompson, C. Bregler, and K. P. Murphy, "Towards accurate multi-person pose estimation in the wild," *CoRR*, vol. abs/1701.01779, 2017. [Online]. Available: http://arxiv.org/abs/1701.01779

[58] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection," *CoRR*, vol. abs/1612.03144, 2016. [Online]. Available: http://arxiv.org/abs/1612.03144

[59] T. Kong, A. Yao, Y. Chen, and F. Sun, "Hypernet: Towards accurate region proposal generation and joint object detection," *CoRR*, vol. abs/1604.00600, 2016. [Online]. Available: http://arxiv.org/abs/1604.00600

[60] K. Su, D. Yu, Z. Xu, X. Geng, and C. Wang, "Multi-person pose estimation with enhanced channel-wise and spatial information," *CoRR*, vol. abs/1905.03466, 2019. [Online]. Available: http://arxiv.org/abs/1905.03466

[61] W. Li, Z. Wang, B. Yin, Q. Peng, Y. Du, T. Xiao, G. Yu, H. Lu, Y. Wei, and J. Sun, "Rethinking on multi-stage networks for human pose estimation," *CoRR*, vol. abs/1901.00148, 2019. [Online]. Available: http://arxiv.org/abs/1901.00148

[62] Y. Kim, C. Denton, L. Hoang, and A. M. Rush, "Structured attention networks," *CoRR*, vol. abs/1702.00887, 2017. [Online]. Available: http://arxiv.org/abs/1702.00887

[63] S. Yang, Z. Quan, M. Nie, and W. Yang, "Transpose: Towards explainable human pose estimation by transformer," *CoRR*, vol. abs/2012.14214, 2020. [Online]. Available: https://arxiv.org/abs/2012.14214

[64] E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, and B. Schiele, "Deepercut: A deeper, stronger, and faster multi-person pose estimation model," *CoRR*, vol. abs/1605.03170, 2016. [Online]. Available: http://arxiv.org/abs/1605.03170

[65] Z. Cao, G. Hidalgo, T. Simon, S. Wei, and Y. Sheikh, "Openpose: Realtime multi-person 2d pose estimation using part affinity fields," *CoRR*, vol. abs/1812.08008, 2018. [Online]. Available: http://arxiv.org/abs/1812.08008

[66] X. Zhu and Y. Jiang, "Multi-person pose estimation for posetrack with enhanced part affinity fields," 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:52563463

[67] G. Hidalgo, Y. Raaj, H. Idrees, D. Xiang, H. Joo, T. Simon, and Y. Sheikh, "Single-network whole-body pose estimation," *CoRR*, vol. abs/1909.13423, 2019. [Online]. Available: http://arxiv.org/abs/1909.13423

[68] S. Kreiss, L. Bertoni, and A. Alahi, "Pifpaf: Composite fields for human pose estimation," *CoRR*, vol. abs/1903.06593, 2019. [Online]. Available: http://arxiv.org/abs/1903.06593

[69] A. Newell and J. Deng, "Associative embedding: End-to-end learning for joint detection and grouping," *CoRR*, vol. abs/1611.05424, 2016. [Online]. Available: http://arxiv.org/abs/1611.05424

[70] Z. Luo, Z. Wang, Y. Huang, T. Tan, and E. Zhou, "Rethinking the heatmap regression for bottom-up human pose estimation," *CoRR*, vol. abs/2012.15175, 2020. [Online]. Available: https://arxiv.org/abs/2012.15175

[71] V. Ferrari, M. Marin-Jimenez, and A. Zisserman, "Progressive search space reduction for human pose estimation," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[72] Y. Yang and D. Ramanan, "Articulated human detection with flexible mixtures of parts," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 35, no. 12, pp. 2878–2890, 2013.

[73] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.

[74] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," *CoRR*, vol. abs/1604.01685, 2016. [Online]. Available: http://arxiv.org/abs/1604.01685

[75] R. Tylecek and R. Sára, "Spatial pattern templates for recognition of objects with regular structure," in *German Conference on Pattern Recognition*, 2013. [Online]. Available: https://api.semanticscholar.org/CorpusID:6060524

[76] P. Isola, J. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *CoRR*, vol. abs/1611.07004, 2016. [Online]. Available: http://arxiv.org/abs/1611.07004

[77] A. Yu and K. Grauman, "Fine-grained visual comparisons with local learning," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 192–199.

[78] J. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros, "Generative visual manipulation on the natural image manifold," *CoRR*, vol. abs/1609.03552, 2016. [Online]. Available: http://arxiv.org/abs/1609.03552

[79] Y. Choi, Y. Uh, J. Yoo, and J. Ha, "Stargan v2: Diverse image synthesis for multiple domains," *CoRR*, vol. abs/1912.01865, 2019. [Online]. Available: http://arxiv.org/abs/1912.01865

[80] P.-Y. Laffont, Z. Ren, X. Tao, C. Qian, and J. Hays, "Transient attributes for high-level understanding and editing of outdoor scenes," *ACM Transactions on Graphics (proceedings of SIGGRAPH)*, vol. 33, no. 4, 2014.

[81] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015.

[82] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *CoRR*, vol. abs/1502.03167, 2015. [Online]. Available: http://arxiv.org/abs/1502.03167

[83] S.-C. Zhu, X. Liu, and Y. N. Wu, "Exploring texture ensembles by efficient markov chain monte carlo-toward a 'trichromacy' theory of texture," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, pp. 554–569, 2000. [Online]. Available: https://api.semanticscholar.org/CorpusID:3194236

[84] V. Dumoulin, J. Shlens, and M. Kudlur, "A learned representation for artistic style," *CoRR*, vol. abs/1610.07629, 2016. [Online]. Available: http://arxiv.org/abs/1610.07629

[85] X. Huang and S. J. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," *CoRR*, vol. abs/1703.06868, 2017. [Online]. Available: http://arxiv.org/abs/1703.06868

[86] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2016.

[87] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *CoRR*, vol. abs/1505.04597, 2015. [Online]. Available: http://arxiv.org/abs/1505.04597

[88] Z. Yi, H. Zhang, P. Tan, and M. Gong, "Dualgan: Unsupervised dual learning for image-to-image translation," *CoRR*, vol. abs/1704.02510, 2017. [Online]. Available: http://arxiv.org/abs/1704.02510

[89] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2022.

[90] H. Hoyez, C. Schockaert, J. Rambach, B. Mirbach, and D. Stricker, "Unsupervised image-to-image translation: A review," *Sensors*, vol. 22, no. 21, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/21/8540

[91] T. Salimans, I. J. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," *CoRR*, vol. abs/1606.03498, 2016. [Online]. Available: http://arxiv.org/abs/1606.03498

[92] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," *CoRR*, vol. abs/1512.00567, 2015. [Online]. Available: http://arxiv.org/abs/1512.00567

[93] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, G. Klambauer, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a nash equilibrium," *CoRR*, vol. abs/1706.08500, 2017. [Online]. Available: http://arxiv.org/abs/1706.08500

[94] M. Fréchet, "Sur la distance de deux lois de probabilité," *Annales de l'ISUP*, vol. VI, no. 3, pp. 183–198, 1957. [Online]. Available: https://hal.science/hal-04093677

[95] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," *CoRR*, vol. abs/1801.03924, 2018. [Online]. Available: http://arxiv.org/abs/1801.03924

[96] H. Xu, J. Wang, X.-S. Hua, and S. Li, "Image search by concept map," in *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '10. New York, NY, USA: Association for Computing Machinery, 2010, p. 275–282. [Online]. Available: https://doi.org/10.1145/1835449.1835497

[97] J. Wang and X. Hua, "Interactive image search by color map," *ACM Trans. Intell. Syst. Technol.*, vol. 3, pp. 12:1–12:23, 2011. [Online]. Available: https://api.semanticscholar.org/CorpusID:6538567

[98] Y. Cao, H. Wang, C. Wang, Z. Li, L. Zhang, and L. Zhang, "Mindfinder: interactive sketch-based image search on millions of images," 10 2010, pp. 1605–1608.

[99] F. Radenovic, G. Tolias, and O. Chum, "Fine-tuning CNN image retrieval with no human annotation," *CoRR*, vol. abs/1711.02512, 2017. [Online]. Available: http://arxiv.org/abs/1711.02512

[100] D. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, 1999, pp. 1150–1157 vol.2.

[101] D. Kadish, S. Risi, and A. S. Løvlie, "Improving object detection in art images using only style transfer," *CoRR*, vol. abs/2102.06529, 2021. [Online]. Available: https://arxiv.org/abs/2102.06529

[102] P. Madhu, A. Villar-Corrales, R. Kosti, T. Bendschus, C. Reinhardt, P. Bell, A. K. Maier, and V. Christlein, "Enhancing human pose estimation in ancient vase paintings via perceptually-grounded style transfer learning," *CoRR*, vol. abs/2012.05616, 2020. [Online]. Available: https://arxiv.org/abs/2012.05616

[103] T. Jenícek and O. Chum, "Linking art through human poses," *CoRR*, vol. abs/1907.03537, 2019. [Online]. Available: http://arxiv.org/abs/1907.03537

[104] E. Ioannou and S. Maddock, "Evaluation in neural style transfer: A review," 2024.

[105] Roman. (2023) Image similarity comparison using vgg16 deep learning model. [Online]. Available: https://medium.com/@developerRegmi/image-similarity-comparison-using-vgg16-deep-learning-model-a663a411cd24

[106] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Knowledge Discovery and Data Mining*, 1996. [Online]. Available: https://api.semanticscholar.org/CorpusID:355163

[107] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLO," Jan. 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[108] H. Chen, F. Shao, X. Chai, Y. Gu, Q. Jiang, X. Meng, and Y.-S. Ho, "Quality evaluation of arbitrary style transfer: Subjective study and objective metric," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 7, p. 3055–3070, Jul. 2023. [Online]. Available: http://dx.doi.org/10.1109/TCSVT.2022.3231041

[109] Y. Xu, J. Zhang, Q. Zhang, and D. Tao, "Vitpose++: Vision transformer for generic body pose estimation," 2023.

[110] F. Zhang, X. Zhu, H. Dai, M. Ye, and C. Zhu, "Distribution-aware coordinate representation for human pose estimation," *CoRR*, vol. abs/1910.06278, 2019. [Online]. Available: http://arxiv.org/abs/1910.06278

[111] J. Huang, Z. Zhu, F. Guo, and G. Huang, "The devil is in the details: Delving into unbiased data processing for human pose estimation," *CoRR*, vol. abs/1911.07524, 2019. [Online]. Available: http://arxiv.org/abs/1911.07524

[112] Z. Geng, K. Sun, B. Xiao, Z. Zhang, and J. Wang, "Bottom-up human pose estimation via disentangled keypoint regression," *CoRR*, vol. abs/2104.02300, 2021. [Online]. Available: https://arxiv.org/abs/2104.02300

[113] W. J. McNally, K. Vats, A. Wong, and J. McPhee, "Rethinking keypoint representations: Modeling keypoints and poses as objects for multi-person human pose estimation," *CoRR*, vol. abs/2111.08557, 2021. [Online]. Available: https://arxiv.org/abs/2111.08557

[114] S. Huang, J. An, D. Wei, J. Luo, and H. Pfister, "Quantart: Quantizing image style transfer towards high visual fidelity," 2023.

[115] Y. Zhang, F. Tang, W. Dong, H. Huang, C. Ma, T.-Y. Lee, and C. Xu, "A unified arbitrary style transfer framework via adaptive contrastive learning," 2023.

[116] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *CoRR*, vol. abs/2010.11929, 2020. [Online]. Available: https://arxiv.org/abs/2010.11929

[117] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M. Yang, "Universal style transfer via feature transforms," *CoRR*, vol. abs/1705.08086, 2017. [Online]. Available: http://arxiv.org/abs/1705.08086

[118] X. Li, S. Liu, J. Kautz, and M. Yang, "Learning linear transformations for fast arbitrary style transfer," *CoRR*, vol. abs/1808.04537, 2018. [Online]. Available: http://arxiv.org/abs/1808.04537

[119] D. Y. Park and K. H. Lee, "Arbitrary style transfer with style-attentional networks," *CoRR*, vol. abs/1812.02342, 2018. [Online]. Available: http://arxiv.org/abs/1812.02342