V4Design

Visual and textual content re-purposing FOR(4) architecture, Design and virtual reality games

H2020-779962

D3.2

Basic version of aesthetics concept extraction algorithms & tools

Dissemination level:	Public
Contractual date of delivery:	Month 16, 30 April 2019
Actual date of delivery:	Month 17, 02 May 2019
Workpackage:	WP3 Visual and textual content analysis
Task:	T3.5 Aesthetic concept and attributes extraction from visual content
Type:	Report
Approval Status:	Draft
Version:	1.0
Number of pages:	42
Filename:	D3.2-V4Design_Basic version of aesthetics concept extraction algorithms & tools_2019-05-02_v1.0.docx

Abstract

Basic version of aesthetics concept extraction algorithms & tools will be tested against the SoA on benchmark data which will include the first iterations of: (a) aesthetics feature extraction, (b) emotional reaction prediction on art image database.

The information in this document reflects only the author's views and the European Community is not liable for any use that may be made of the information contained therein. The information in this document is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

co-funded by the European Union

History

Version	Date	Reason	Revised by
0.1	27-02-2019	Table of content created & content defined	K. Avgerinakis
0.2	29-03-2019	Initial version and assignments distribution	E. Batziou
0.3	15-04-2019	Merge contributions and draft initial version	E. Batziou
0.4	22-04-2019	Revision and update of the introduction and conclusion sections	K. Avgerinakis
0.5	25-04-2019	Integration of different section and finalization of the document	E. Batziou
0.6	30-04-2019	Internal review of the 1st version	Jens Derdaele
1.0	02-05-2019	Version addressing internal review by KUL	K. Avgerinakis

Author list

Organisation	Name	Contact Information
CERTH	Elissavet Batziou	batziou.el@iti.gr
CERTH	Konstantinos Avgerinakis	koafgeri@iti.gr
CERTH	Stefanos Vrochidis	stefanos@iti.gr

Executive Summary

This deliverable reports on the basic techniques for aesthetics concept extraction and texture proposals. Specifically, methods for the analysis of aesthetics (paintings, pictures of artwork, pictures of interiors and exteriors of buildings) and multimedia data are profoundly described in this document. The goal of aesthetic extraction is to extract concept and conceptual relations amongst the acquired visual information, while methods for texture proposal focus on the detection of concepts from visual content.

The document describes in detail the WP3 modules, which are related to T3.5 and the appropriate approaches, components, and resources that were adopted so as to accomplish the respective functionalities that were described in the DoA and later on the ones that documented from the users throughout the compiled user requirements (D7.1, D7.2). The deliverable introduces the basic techniques for aesthetics concept extraction and texture proposals that were deployed during the first phase of the project's lifetime, for the implementation of the 1st prototype (M18). Furthermore, a description of the analysis requirements for visual content are provided and analysed appropriately. While, for each module an overview of the State-of-the-Art (SoA) and a comparison to other approaches is included. The evaluation approaches and results are finally explained and demonstrated at the end of the document.

More specifically, the following modules are described in further details:

a) The Aesthetics Extraction (AE) module from visual content (image/video), which includes the recognition of the style, creator, genre of famous paintings and architectural buildings.
b) The Texture Proposal (TP) module, which uses aesthetics extraction output so as to transfer its style to textures and use it for changing 3D models of architectural buildings.

Aesthetics Extraction is the responsible module for recognizing new images and paintings and integrate them in V4Design aesthetics database, while Texture Proposal use these models to provide novel textures to V4Design users (i.e. architects, video game designers).

CERTH was responsible for the development of the described methodologies and modules for aesthetics recognition and texture proposals (Task 3.5). While, it is worthwhile to note, that the performance of the above modules was extensively evaluated in terms of classification and recognition accuracy, as well as user likeness for texture proposals. The first experimental results were far from encouraging to continue working towards this scientific direction.

Abbreviations and Acronyms

AE	Aesthetics Extraction
AdaIn	Adaptive Instance normalisation
AVA	Aesthetics Visual Analysis
BMP	Bitmap
CAN	Creative Adversarial Network
CH	Could Have
CNN	Convolutional Neural Network
DCNN	Deep Convolutional Neural Network
DcCNN	Deep Columnar Convolutional Neural Network
DMA	Deep multi-patch aggregation
DoA	Description of Actions
EAEF	Elements of Art based Emotion Features
FBX	Filmbox
FC	Fully Connected
GAN	Generative Adversarial Network
GIST	Global structural features
GPU	Graphics Processing Unit
HSV	Hue Saturation Value
IoU	Intersection over union
JPG	Joint Photographic Group
MH	Must Have
MSE	Mean Square Error
NiN	Network in Network
NN	Neural Networks
PAEF	Principles of Art based Emotion Features
PCA	Principal Components Analysis
PCC	Pearson Correlation Coefficient
PNG	Portable Network Graphics
PUC	Pilot Use Case
RDCNN	Regularised Deep Convolutional Neural Network
ReLU	Rectified Linear Units
RGB	Red Blue Green
RR	Recognition Rate
SCNN	Single-column Convolutional Neural Network
SGD	Stochastic Gradient Descent
SH	Should Have
SIFT	Scale Invariant Feature Transform
SoA	State of the Art

SVM	Support Vector Machine
TIFF	Tagged Image File Format
TP	Texture proposal
UR	User Requirement
VGG	Visual Geometry Group
Vrmat	Vray Materials
WCT	Whitening Colouring Transformation
WH	Won’t Have
ZCA	Zero phase Component Analysis

Table of Contents

1 INTRODUCTION

In V4Design, the role of T3.5 (Aesthetic concept and attributes extraction from visual content) is to extract the aesthetics concept out of images of architecture and paintings, as well as videos and movies in order to create an annotated aesthetics database that will be used in order to create novel texture proposals for changing the 3D-models, which will be created through T4.3-T4.4, and will be provided as an alternative to V4Design end-users.

During the first half of V4Design project lifetime (M1-M16), T3.5 contributed to the 3rd Milestone MS3 "1st Prototype and evaluation" for the successful completion of the first SW development cycle of the project as shown in Figure 1. Generally, the objective of T3.5 was to provide the appropriate technological tools that will allow V4Design platform to: (i) Extract the aesthetics from the compiled visual content, such as paintings and images of architectural structures, and (ii) Propose novel textures from the recognized and formulated aesthetics so as to change the extracted 3D models .

	2018												2019												2020											
Tasks	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
WP3. Visual and textual content analysis	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■			
T3.1: Compilation and study of texts relevant to visual data	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩																								
T3.2: Entity identification and linking, word sense disambiguation and lexical modeling			🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩			
T3.3: Dependency-based semantic parsing			🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩			
T3.4: Conceptual relation extraction					🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩			
T3.5: Aesthetic concept and attributes extraction from visual content			🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩			

Figure 1. WP3 tasks towards V4Design lifetime.

Task 3.5 interacts both internally with WP3 tasks and more specifically with T3.4, which is the main task that accumulates textual information and can tag the visual data analyzed from aesthetics extraction, but also interacts with tasks from external WPs, such as WP2, WP4, WP5, WP6 and WP7. Initially, T3.5 is closely related to T2.1, T2.3 and T2.4, which are the responsible tasks to accumulate visual data of architectural structures and paintings and create an annotated corpus so as to train the AE & TP modules. T3.5 provide its outcome to T4.4 in order to texturize the reconstructed 3D models acquired from T4.3 and T5.1 and T5.2 in order to populate V4Design Knowledge Base (KB). T3.5 is also related to T6.1, T7.1, T7.2, where the user and technical requirements are defined, and T6.4 where service integration is performed. Thus, it is obvious that it is an essential and useful service for V4Design platform, intercorrelated with several tasks of the overall Description of Actions (DoA).

1.1 Objectives

The objectives of Tasks 3.5 for the 1st period of the project (M1-M16) are in aligned with the main goals, as they were described in the DoA, and summarised to the following:

- Analyse visual content from paintings and images of other kind of artwork in order to extract geometry, style and other aesthetics aspects concerning specific artwork collections.
- Provide the outcome of aesthetics concept extraction as metadata to V4Design platform, and artwork features to be extracted will be specified in measurable attributes, such as colour (RGB, HSV, etc.), texture, image bumps, gradients, palettes, and patterns.

- Extract discriminative and compact aesthetics features from famous paintings so as to understand, represent and categorize their creator, style and genre.
- Analyse images of architectural material and structures in order to deploy a recognition framework that will be able to understand who and when has created the depicted artefacts.
- Analyse the spatio-temporal visual features that exist in the sequential order of video frames and the accompanied audio in order to extract video aesthetics and the sentiment that it produces to its viewers.
- Build an aesthetics visual "bank" of styles, genres and creators that will be used in order to create the texture proposals framework.
- Transfer the pattern of landscape images and famous paintings to target images of video frames so as to be used for retexturizing 3D models.

1.2 Results towards the foreseen objectives of V4Design project

Until now, V4Design has fulfilled the foreseen objectives of the project by completing the development of the basic functionalities of aesthetics extraction and texture proposals with the following activities:

a) Accumulated annotated visual data from benchmark datasets (WikiArt, Pandora, Paintings-91) and V4Design consortium partners (IC029, IC032 as reported in D4.1)
b) Deployed the initial version of Aesthetics Extraction (AE) in paintings by deploying State of the Art (SoA) computer vision and deep learning algorithms in the compiled datasets.
c) Deployed the initial version of Texture Proposals (TP) in images and video samples by transferring paintings aesthetics style in target architectural images.
d) Developed the Aesthetics that movies may invoke to viewers by deploying a spatio-temporal deep learning representation that leverages audio-visual features to understand what happen in the sequential video scenes.

1.3 Future plans

As far as the advanced techniques for Aesthetics Extraction and Texture Proposals are concerned, V4Design plan to deploy the following activities until the end of V4Design project.

a) Create aesthetics extraction clusters so that style can be transferred not only from single images but from cluster of images as well. This will help to the creation of the style transfer that will be provided by V4Design plugin.
b) Transfer styles from paintings and other artistic material using a weighted heatmap so as to texturize 3D models harmonically. In other words, use weighted segmentation masks that will be transfer background and foreground features to the background and the foreground of the target image respectively.
c) Recognize and transfer aesthetics from movies (i.e. horror, comedy architecture etc)
d) Deploy computer vision algorithms that will leverage architectural features so as to understand the style and creator of architectural artefacts.

1.4 Outline

The outline of this deliverable includes a briefly presentation of user requirements for the analysis of the visual and audio content for the aesthetics extraction and texture proposals modules, as well as a description of the state-of-the-art methodologies in the scientific fields of computer vision, deep learning and style transfer. The methodology analysis of the two modules are then described in Sections 4 and 5, and evaluated in together with their comparison to SoA Section 6. Section 7 concludes the deliverable with foreseen steps of AE and TP until M33.

2 AESTHETICS EXTRACTION REQUIREMENTS

The V4Design user requirements have been identified in D7.2 "Initial use case scenarios and user requirements". Some of them are associated and directly linked to Aesthetics Extraction and Texture Proposals, as detailed in Section 2.1 and Section 2.2 . It is worthwhile to note that as far as Aesthetic Extraction (AE) and Texture Proposal (TP) components are concerned, CERTH not only took into account the following user requirements (Sec2.1, Sec2.2) so as to produce the initial functionalities of V4Design system, but also augment their metadata output so that the users can receive extra AE&TP functionalities and reiterate the discussion between the user and the technical partners.

2.1 Aesthetics concept extraction requirements

Two user requirements from D7.2 have been associated with the aesthetics extraction module of V4Design, namely the UR_41 and UR_42. Regarding UR_41, an architect wants to recognise texture and material from images and videos. Aesthetics Extraction classifies visual content with respect to their style and aesthetic characteristics. In the UR_42, an architect wants to have the "intelligence" of an architectural synthetic tool (combination of texture, colours, shapes). Aesthetics Extraction will accomplish these requirements by identifying combinations of texture, colours and shapes of visual data to categorise them into classes by their school of art or creator, using computer vision and deep learning techniques.

User Requirement (UR)	Associated HLUR	Detailed description	Functional or Non Functional (FR/N-FR)	Priority based on MoSCoW framework
UR_41	HLUR_203 HLUR_205 HLUR_206	As an Architect I want texture and material recognition that might appear in images and videos.	N-FR	CH
UR_42	HLUR_203 HLUR_208	As an Architect I want to have the "intelligence" of an architectural composition tool (combination of texture, colours, shapes)	N-FR	CH

Table 1: Relevant user requirements reported in D7.2 for Aesthetic Extraction

2.2 Texture proposals requirements

Six user requirements from D7.2 have been associated with the texture proposals module of V4Design, namely the UR_3, UR_4, UR_27, UR_37, UR_42 and UR_43. In the UR_3, an architect wants to be able to retrieve high and reduced resolution textures. Texture Proposal (TP) module reduces texture resolution at the half of the original input image and can provide a high-quality texture proposal back to the user, as required. Regarding UR_4, an architect wants to be able to reuse textures (Pattern extraction / seamless texture generation). Texture Proposal is able to save the produced visual content in order to be reused and provide it back to the users. In UR_27 a user wants various file formats as outputs and more specifically the compressed image files JPG, TIFF, BMP and PNG for texture proposals. TP produces outputs with the same file formats as the one's requested by the users by leveraging FFMEG functionalities, which is freely available in OPENCV and on the web. As far as UR_37 and UR_43 is concerned, the TP module will provide to the system a set of available texture proposals so as to visualize them in the authoring tools (UR_37) and the Virtual Reality environment (UR_43) to the users and let them choose from a predefined set of styles. Finally, in the UR_42, an architect wants to have the "intelligence" of an architectural synthetic tool (combination of texture, colours, shapes). Texture Proposal module identifies combinations of texture, colours and shapes of visual data to pass them in the new produced visual content.

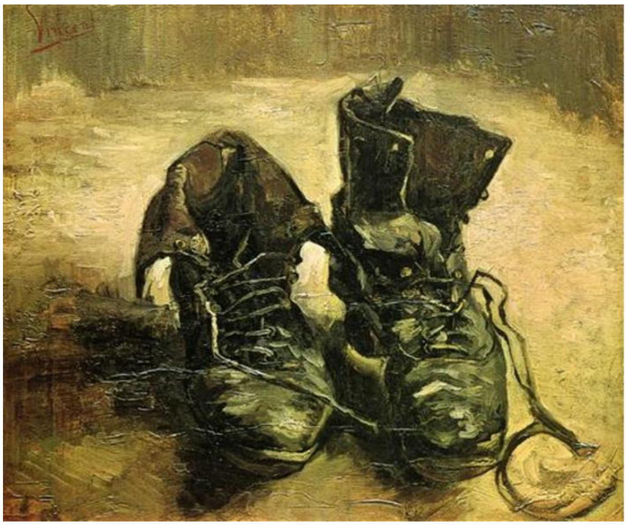
User Requirement (UR)	**Associated HLUR**	**Detailed description**	**Functional or Non Functional (FR/N-FR)**	**Priority based on MoSCoW framework**
UR_3	HLUR_202	As an Architect I want to be able to retrieve high and reduced resolution textures	FR	MH
UR_4	HLUR_202	As an Architect I want to be able to reuse textures (Pattern extraction / seamless texture generation)	FR	CH
UR_27	HLUR_204	As a user I want various output file formats such as JPG, TIFF, BMP and PNG for textures	FR	SH
UR_37	HLUR_203 HLUR_207	As an Architect I want UIX: Detailed search by features: - Quality (3D model/ texture), Footage features, augmented data	N-FR	SH
UR_42	HLUR_203 HLUR_208	As an Architect I want to have the "intelligence" of an architectural composition tool (combination of texture, colours, shapes)	N-FR	CH
UR_43	HLUR_203 HLUR_207	As an Architect I want to browse assets (materials, textures, bumps etc.) in VR/AR environment (not only in screen)	N-FR	CH

Table 2: Relevant user requirements reported in D7.2 for Texture Proposals

3 RELEVANT WORK

3.1 Aesthetics extraction

One of the most intriguing domains in architecture, design and video game creation is the extraction of aesthetics and style from paintings, movies, artwork artefacts, buildings and more generally outdoor spaces, such as squares and large architecture structures. Artwork artefacts, such as paintings, have style which characterises the school of art that it belongs to and the artist who created it, as demonstrated in **Figure 2**. Style is a term which refers to several aspects of art, such as the techniques used to create a painting, the philosophy behind the painting, or the form of expression employed by the creator. A school of art is a group of creators who have been influenced from the same teachers or they share a common style, theme or ideology. To distinguish to which school of art belongs a painting, or who is its creator is a challenging problem for a human non-expert.



Style: Post Impressionism

Genre: Still life

Artist: Vincent Van Gogh

Figure 2: A pair of shoes- Vincent Van Gogh, 1886

3.1.1 Aesthetics extraction from paintings

Earlier approaches in the wider sense of aesthetics in an image mainly focus on the aesthetic quality assessment, where low-level features are extracted to classify an image in terms of quality. Initially, the Bag-of-colour-patterns model has been introduced in (Nishiyama, 2011) to classify photos with respect to colour harmony. A photo is described by a collection of local regions, then, hue, chroma and brightness values are calculated for all pixels within each region, and these colour features are quantised as a histogram. Another approach on the aesthetic quality of paintings is evaluated in terms of a group of novel low-level features in (Li C. &., 2009). These features are based on global and local characteristics. Global characteristics involve hue, saturation, lightness (value), brightness features, blurring effect and edges distribution. Local features involve shape of segments, colour features of segments, contrast features between segments and a focus region based on rules.

The aim of authors in (Elgammal A. &., 2015) is to quantify creativity and influence in networks of paintings. Each painting is a node and the link from painting i to painting j is weighted by the similarity between i and j. Creativity score is computed using a modified

PageRank centrality measure on the weighted network of paintings. The features which are used to represent each painting are the low-level GIST visual descriptors (Oliva, 2005). The above methodologies have been significant contributions in aesthetic quality assessment and could be examined in the aesthetics and style classification too. Moreover, Saleh and Elgammal in (Saleh, 2015), explore a variety of features and metric learning approaches for computing the similarity between paintings and styles.

High-level features have demonstrated their effectiveness in various image classification tasks, such as image aesthetics categorisation. Several authors have developed systems to classify classic painting styles. Shamir et al., in (Shamir, 2010) firstly consider a dataset which includes painting images of three styles (impressionism, expressionism and surrealism). Each style is represented by three artists. 57 images are collected for each artist, and the split is done into a training set of 40 images and a test set of 17 images. They are based on a set of features and extend it by using combinations of them. Moreover, a Fisher score calculated for all considered features and also a similarity score is proposed for painting-to-class similarity. In (Jain, 2019) a multi-layer ``deep'' neural network is introduced to learn features trained on object class categories (ImageNet (Krizhevsky, 2012)). The authors use the ImageNet eight-layer convolutional network, trained on over a million images annotated with 1,000 ImageNet classes. Experiments on two image collections (photos, paintings) show that the proposed features outperform state-of-the-art visual features such as colour histogram and GIST. A different Neural Network approach is introduced in (Lu X. L., 2014) to categorise images with respect to aesthetic features. Firstly, a Single-column Convolutional Neural Network (SCNN) is constructed using as input three global views of an image and one local random crop view. Then a Double-column CNN (DcCNN) is proposed where columns in different columns are independent in convolutional layers and the first two fully connected layers, while the final fully connected layers are jointly trained. The Regularised DCNN (RDCNN) springs from the combination of SCNN and DcCNN in the AVA dataset (Murray, 2012), which involves style and aesthetic ground-truth labels. Later, in (Lu X. L., 2015) the authors introduce a deep multi-patch aggregation network architecture (DMA-Net) to learn and categorise images using style attributes in aesthetic quality assessment. The input is a bag of five patches which is formulated by five multiple random local crops of an image. The aggregation structure is either Statistics layer or a Fully-Connected Sorting layer. This approach introduces novel DCNN-features that can be involved in aesthetics extraction and style classification. In (Elgammal A. L., 2017) the authors propose a method based on Generative Adversarial Networks (GAN) named Creative Adversarial Networks (CAN) in which the system generates creative art by looking at art and learning about style. The proposed network has a generator which generates art randomly and a discriminator which labels the generated art as art or not and classifies it by art style (e.g. Impressionism, Baroque, Cubism, etc.).

3.1.2 Aesthetics extraction in images of architectural buildings

There are also some approaches relevant to aesthetics extraction in buildings images, as it is described in the following. Scale Invariant Feature Transform (SIFT) descriptors are involved in (Shalunts, 2011), where a method is proposed to classify facade windows by their architectural style. Their approach based on learning of local features and Scale Invariant Feature Transform (SIFT) descriptors and clustering them to learn a visual vocabulary. Other factors could be associated with the aesthetics of a building, such as its age. In (Zeppelzauer,

2018) a method for automatic age estimation of buildings photographs is proposed. The approach firstly learns characteristic visual patterns for each building epochs based on the most promising image patches from the input images, then classifies them using pre-trained CNN at different object types and finally estimate the building age.

3.1.3 Aesthetics extraction in movies

An intriguing trend that appears to get a lot of attention lately is recognizing the emotion from some painting or some specific section of a movie, recognising it and its impact to the viewer.

In (Zhou B. L., 2014) the authors describe elements-of-art based emotion features (EAEF), such as colour, value, line, texture, shape, form and space and introduce principles-of-art based emotion features (PAEF), such as balance, emphasis, harmony, variety, gradation and movement. They apply the proposed PAEF to predict the emotions implied in famous artworks using SVM classification. In (Baveye, 2015) the authors introduce a new dataset composed of 30 movies annotated along the induced valence and arousal axes. Moreover, the authors found that the fine-tuned CNN framework is a promising solution for emotion prediction. State-of-the-art approaches in emotion recognition from visual content have mainly focused in techniques which are based on Convolutional Neural Networks (CNNs). In particular, the authors in (Li S. D., 2017) propose a Deep Locality-Preserving CNN method, aiming to enhance the discriminative power of deep features in order to study the common expression perception (e.g. smile vs laugh), based on a real-world publicly available facial expression database RAF-DB (9,672 real-world images labeled for different expressions). In (Guo, 2018) the authors train multiple CNN models to learn high-level abstractions of the input from different perspectives. The trained models are fused into a high-performance hybrid network, which focuses on faces, scenes, skeletons (face, pose, and hand) and regions extracted with visual attention mechanisms for group-level emotion recognition. The main objective of (Zhang, 2018) is to bridge the emotional gap between visual and audio content, using a hybrid deep model. It first produces audio–visual segment features with CNNs and 3D-CNN, and then fuses them into audio–visual segment features in Deep Belief Networks (DBNs). The method is tested in three datasets (RML, eNTERFACE05, and BAUM-1s datasets) which all include the six basic emotions joy, anger, sadness, disgust, fear, surprise. Jain et al, in (Jain, 2019) introduced a deep learning model, combining a Fully Connected Network and a residual block, for emotional recognition, which learns the subtle features that discriminate the different facial expressions (sad, happy, surprise, angry, neutral, disgust, fear). In the context of V4Design we present in section 3.1.3 our transfer learning approach for emotion recognition in movies.

3.2 Texture proposals

Texture proposal is an extension of aesthetics extraction, in the context of learning the elements of a style and then transferring this style to another image with a specific given content. For example, an image of Gendarmenmarkt square in Berlin would be combined with one of the most famous paintings of Vincent Van Gogh, ``Cafe Terrace at night'', to generate a new image of the Gendarmenmarkt square with the style of Van Gogh, as it is depicted in **Figure 3.** Moreover, the proposed textures augment 3D-models, leading into an enhanced 3D-model with modified aesthetics style information. Texture proposal may also be extended to videos, aiming to change the stylistic elements of it.

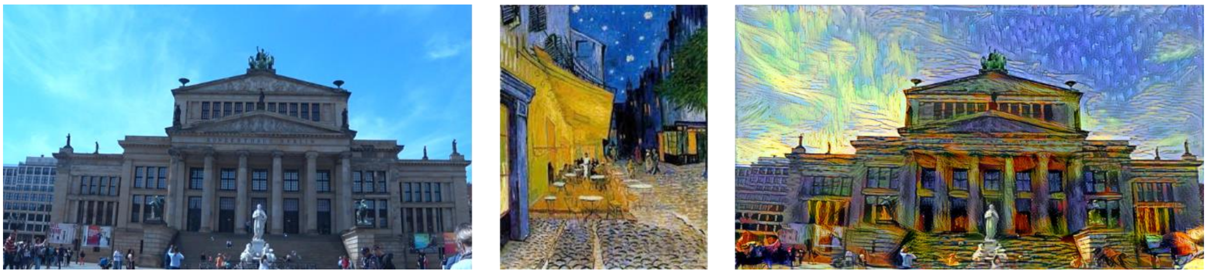


Figure 3: Transferring in an image of Gendarmenmarkt the style of the famous painting Cafe Terrace at Night of Vincent Van Gogh.

3.2.1 Texture proposals from paintings

In texture proposals problem the work of the authors in (Gatys L. A., 2016) have shown that Deep Neural Networks (DNN) encode not only the content but also the style information of an image. The introduced texture proposals method is flexible enough to combine content and style of arbitrary images. The authors for the first time demonstrate impressive texture proposals results by matching feature statistics in convolutional layers of a DNN. Their framework is based on a slow optimisation process that iteratively updates the image to minimise a content loss and a style loss computed by a loss network. This framework matches styles by matching the second-order statistics between feature activations, as they are captured by the Gram matrix. The authors in (Huang, 2017) adjust channel-wise statistics of the content features by adaptive instance normalisation (AdaIN) and train a feature decoder by a combinational scale-adapted content and style losses. AdaIN receives a content input and a style input and simple aligns the channel-wise mean and variance of content to match those of style. Moreover, AdaIN has no learnable affine parameters. The authors adopt an encoder-decoder architecture in which the encoder is fixed to the first few layers of a pre-trained VGG-19. After encoding the content and style images in feature space, they feed both feature maps to an AdaIN layer that aligns the mean and variance of the content feature maps to those of the style feature maps, producing the target feature maps. In (Ulyanov, 2017), the authors introduce an instance normalisation module to replace batch normalisation in order to improve the performance of the deep neural generators in image stylisation problem. Then, in order to improve the diversity, they introduce a new learning formulation that encourages generators to sample by texture networks. These two ways take feed forward texture synthesis and image stylisation closer to the quality of generation via optimisation while retaining the speed advantage.

Recently, in (Sheng, 2018) the authors proposed feature decoration that generalises AdaIN and whitening and colouring transformation (WCT). Moreover, they use Zero-phase Component Analysis (ZCA) operation in their style transfer method. The VGG network is utilised to extract image features, and ZCA is used to project features into the same space. Then transferred features are obtained by a reassembling operation based on patches. Finally, the transferred features and a decoder network are trained by MSCOCO dataset and utilised to reconstruct the styled image. In (Xu, 2018), the authors trained adversarially a feed-forward network for arbitrary style transfer. They introduce techniques to tackle the problem of adversarial training from multi-domain data. In adversarial training the generator (stylisation network) and the discriminator are alternatively updated. Both of them are conditional networks; the generator is trained to fool the discriminator, as well as satisfy the content and style representation similarity to inputs. The generator is built upon the previous work (Huang, 2017) for arbitrary style transfer and the discriminator is conditioned on the coarse domain categories, which are trained to distinguish the generated images from the same style category. Moreover, they propose a mask module to automatically control the level of stylisation by predicting a mask to blend the stylised features and the content features. Finally, they use trained discriminator to rank and find the representative generated images in each style category.

3.3 Summary

We have presented state-of-the-art works in AE for paintings images, images of architectural buildings and videos from movies. Moreover, TP methodologies have been described aiming to present state-of-the-art approaches for generating new images and videos from given visual content and style. We observe that most of the recent works are based on Neural Network architectures. In the following, we present our developed methodologies and we evaluate them with respect to benchmark data collections.

4 AESTHETICS CONCEPT EXTRACTION V1

4.1 Aesthetics extraction from painting images

Nowadays, there is a rising interest in perceiving image aesthetics and several works describe methods to predict image style in an automatic way. The expansion of Deep Convolutional Neural Networks (DCNNs) in computer vision has improved the efficiency and effectiveness of the classification of paintings images not only by artist or year, but also by its aesthetics. However, both training and testing phases require scalable approaches, since we are in the Big Data era, and not only effective ones. Pre-trained models may not only offer knowledge from very large annotated image collections, but also contribute to the efficiency of the learning process through transfer learning.

In this work we present a comparison of state-of-the-art methods on large-scale style, genre, and artist classification of fine-art paintings, as demonstrated in **Figure 4**. We use transfer learning from the Places2[1] dataset introduced in (Zhou B. L., 2014), motivated by the performance shown in (Zhou B. L., 2018), especially when combined with VGG16 architecture. Moreover, we observe that most paintings depict some landscape or other abstract elements that can be represented by Places2 in a more efficient way than others. The extraction of style from artwork in an effective way is a challenging problem and the corresponding visual features need to be extracted with specified and measurable attributes, such as colour (RGB, HSV, etc.), texture, image bumps, gradients, palettes, and patterns, based on state-of-the-art features that exploit external knowledge from pre-trained models and deep learning techniques. We propose a framework that efficiently and effectively classifies paintings images by style, genre and creator. This work analyses visual content from paintings images, as representative form of artwork, in order to extract geometry, style, and creator concerning specific artwork collections.

We propose a novel framework in which we include the Combination of VGG16 DCNNs pre-trained on a large collection of places, followed by its fine-tuning. Contrary to the current state-of-the-art approaches which train models from scratch or use images of general interest, our approach involves transfer learning from images (Places[1] dataset) that potentially could inspire a creator of a painting and usually appears in the background of paintings as a base. Furthermore, our framework is also computationally efficient as it trains only the late layers of a Deep CNN and by this way it improves the time than a baseline model, such as AlexNet-FT-4 (Florea, 2017) requires to be trained, by a factor of up to 17.

4.1.1 Background

Deep Convolutional Neural Networks (DCNNs) have two components; one on the hidden layers for the feature extraction part and one for the classification part. In the feature extraction component, the network combines a sequence of convolution and pooling operations where the features are progressively detected. In the classification part, the fully connected layers serve as a classifier on top of these extracted features, assigning a probability for each class that the algorithm predicts.

[1] http://places2.csail.mit.edu

Convolution is one of the main operations in a DCNN architecture, being the mathematical combination of two tensors to produce a third one. The convolution is performed on the input data with the use of a filter (known also as kernel) to then produce a feature map. We execute a convolution by sliding the filter over the input, whether it is a 2D or 3D array of elements. At every location, a matrix element-wise multiplication is performed and sums the result onto the feature map. The output of the convolution is passed through an activation function. Stride is the step the convolution filter displacement for each step and is usually equal to one meaning that the filter slides pixel by pixel.

In general, the size of the feature map is always smaller than the input, hence it is common to prevent the feature map from shrinking using padding.

After one or a stack of convolution layers, it is common to add one pooling layer to continuously reduce the dimensionality, thus to reduce the number of parameters, so as to decrease the training time. The most frequent type of pooling is max pooling, which takes the maximum value in each considered window.

The convolution and pooling layers are then followed by a few fully connected layers (FC), which can only accept one-dimensional data. To convert our 3D feature array to one-dimensional vector we “flatten” the array by concatenating the rows of each dimension. This vector is further passed to a logistic regression classifier to produce the final vector of class score predictions.

4.1.2 Proposed framework

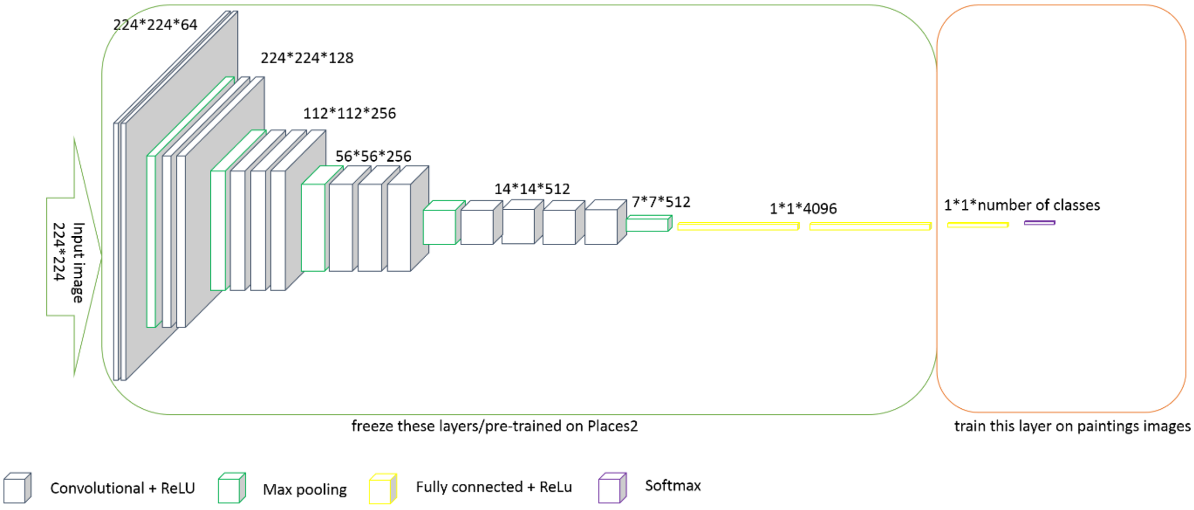


Figure 4: The framework of our AE approach

The input size of our training set is a set of m images with dimensions $n_h * n_c * n_w$, where n_h and n_w are the height and the width of an image with n_c channels. VGG16 is a 16-layer neural network, not including the max-pool layers and the softmax activation in the last layer.

In particular, the image is passed through a stack of convolutional layers, which are used with filters of a small receptive field $f * f$. Spatial pooling is carried out by five max-pooling layers, which follow some of the convolutional layers (not all), as described in the original paper (Simonyan, 2014). Max-pooling is performed over a $w_p * w_p$ pixel window, with stride s. The width of convolutional layers starting from 64 in the first layers and then increasing by a factor of 2 after each max-pooling layer, until reaches 512. The stack of convolutional layers is followed by three Fully-Connected (FC) layers. The final layer is the softmax layer. All hidden layers are equipped with the Rectified Linear unit (ReLU (Krizhevsky, 2012)) which is defined as follows:

$$f(x) = \max(0, x) \tag{1}$$

ReLU is an element-wise operation, applied per pixel, and replaces all negative pixel values in the feature map with zeros. The main property of ReLU is the introduction of non-linearity of the Convolutional Network and therefore, is able to identify and extract realistic non-linearity.

On the FC layers we perform the dropout regularisation in order to reduce complex co-adaptations of neurons and freeze the first fifteen layers of the network which have already been trained on the Places2 benchmark dataset as we can observe in the **Figure 4**. The reason we freeze the weights for the first fifteen layers is, on the one hand, to keep the elements that may be extracted from the background of images that depict landscapes and then train the last layer with paintings features, so that we can have a more artistic representation of the scenes, and on the other hand, to reduce the overall training computational cost.

For the optimisation part of the neural network we adopt the stochastic gradient descent (SGD) method, on the loss function of sparse categorical cross entropy, where the solution is iteratively approached as follows:

$$w := w - \eta \nabla Q_i(w) \tag{2}$$

where η is the learning rate and $Q_i$is the loss of the i example in the training dataset. Notice that we use a learning rate of 0.003, which is smaller than the learning rate for training scratch model (usually 0.01).

Finally, we load each paintings images dataset, split it into training and test sets and starts fine-tuning the model.

4.2 Aesthetics extraction from movies

Emotional Impact of Movies Task[2] was an intriguing challenge of MediaEval 2018 that comprised of two subtasks: (a) Valence/Arousal prediction and (b) Fear prediction from movies. The Task provided a great amount of movies video, their visual and audio features and annotations (Baveye, 2015). Both subtasks ask from the participants to leverage any available technology in order to determine when and whether fear scenes occur and to estimate a valence-arousal score for each video frame in the provided test data. CERTH identified that this dataset can be leveraged by V4Design in order to specify, design and propose textures that could invoke a specific emotion to their viewers. For instance, it could be used by video game designers so as to re-texturize a VR environment and create scarier game scenes, inspired from famous movies and documentaries.

For that purposes, CERTH-ITI introduced its algorithms for valence/arousal and fear recognition subtasks, which included the deployment of deep learning and other

[2] http://www.multimediaeval.org/mediaeval2018/emotionalimpact/

classification schemes to recognise the desired movie style in parts of videos. More specifically, a 3-layer neural network(NN) and a simple linear regression model were deployed, with and without PCA, to predict the correct emotion in the valence-arousal subtask, while a pre-trained VGG16 model (Simonyan, 2014) was fine-tuned, in order to leverage the visual attributes respectively and identify the correct boundary video frames in the fear subtask.

4.2.1 Valence-Arousal Subtask

In the valence-arousal recognition subtask, key-frame extraction was initially applied in order to extract one video frame per second and correlate them with the annotations that were provided from MediaEval benchmark, which was also used the same time interval to record human extracted ground truth data. The provided visual features were then concatenated into one vector representation so as to have a common and fixed representation scheme throughout different video samples.

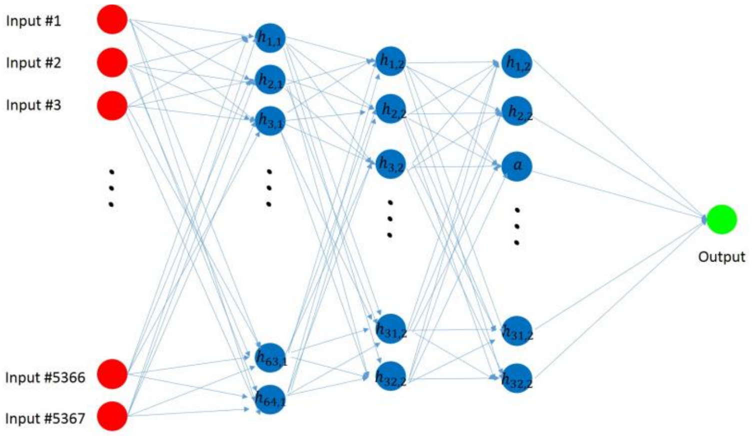


Figure 5: The 3-hidden layer NN of our approach.

The first recognition approach that was deployed concerns the valence/arousal estimation by adopting a linear regression model. Linear regression tried to minimise the residual sum of squares between the ground truth and predicted responses by using linear approximation (Run 3). The concatenated visual features of the provided development set were used to train the model and then the corresponding concatenated visual features of the test set pass through the model in order to predict a score.

PCA is also deployed on our final visual features vectors so as to reduce their dimensionality and keep only the most discriminant principal components (in our case the first 2000) to represent all features (Run 4). On top of this approach, we optionally combined the linear regression model with PCA with 2000 principal components on the concatenated visual features in order to down-sample high dimensional vectors.

A Neural Network (NN) framework was also deployed so as to fulfil the valence/arousal recognition subtask. For that purposes, a 3-hidden layer NN with ReLU activation function and Adam optimiser with learning rate=0.001 was deployed as depicted in **Figure 5**. The size of each hidden layer is 64, 32 and 32 respectively. We use batch size equal to 10 and 10 epochs. The size of the training set is 2/3 of the development set and the remaining 1/3 for validation set. The input of the NN is the set of vectors of concatenated visual features (Run

3). PCA has also been used in order to down-sample the concatenated highly dimensional size (5367) in the golden section of 2000 principal components (Run 4).

4.2.2 Fear Subtask

For the fear recognition subtask, we initially key-frame extraction every one second, as we perform in valance subtask. The frames annotated as "fear" were significantly less than the "no-fear" class and, therefore, in order to balance our dataset, we used data augmentation techniques. Firstly, we downloaded from Flickr about 10,000 images with tag "fear" and we also download emotion images[3] and kept those which are annotated as "fear". In order to further increase the number of fear frames, we additionally use data augmentation techniques on the provided annotated frames. We randomly rotate and translate pictures vertically or horizontally and we randomly apply shearing transformations, randomly zooming inside pictures, flipping half of the images horizontally and filling in newly created pixels which can appear after a rotation or a width/height shift. Finally, we reduce the set of no-fear frames. After these, we had about 23,000 "fear" and 30,000 tagged as "no fear" images to train our model. We used transfer learning to gain information from a large scale dataset and also trained our model in a very realistic and efficient time.

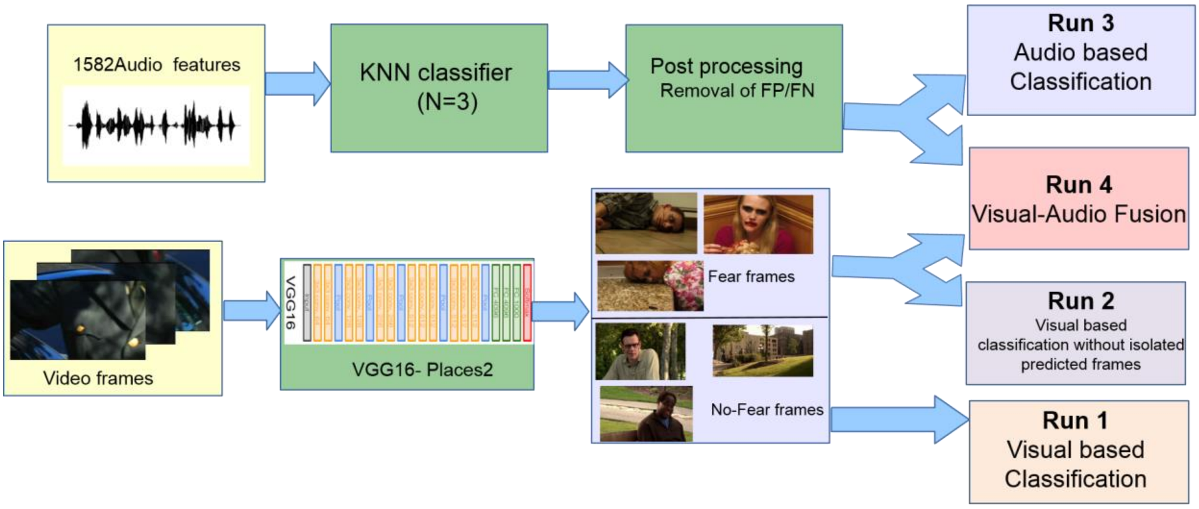


Figure 6: Block diagram of our approach in emotion recognition of movies

The architecture that we chose to represent our features is the VGG16 pre-trained on Places2 dataset (Zhou B. L., 2018) because the majority of the movies have places as background and so we assume that it would be helpful. We use Nadam optimiser with learning rate 0.0001. The batch size is 32 and the number of epochs 50. Finally, we set a threshold of 0.4 on their probability (Run 1). In a different approach, we used the same architecture without isolated predicted frames (Run2).

Additionally, in order to exploit auditory information, we developed a classification method applied on audio features already extracted from the challenge committee using openSmile toolbox (Eyben, 2013). Audio feature vectors, consisting of 1582 features, extracted from videos every second, were separated into training (80%) and validation set (20%). In order

[3] http://www.imageemotion.org/

to equalize the size of the two classes in the training set we randomly removed "no-fear" samples.

We apply KNN classification method with $N = 3$ on the test set, results were further processed, in order to remove erroneous false negatives (single "no-fear" samples around "fear" areas) and false positives (isolated small "fear" areas consisting of one or two "fear" samples).

Results from visual and audio analysis were submitted both separately, as different runs, and in combination by taking the post probabilities of visual and auditory classifications and setting a threshold of 0.7 on their average probability. The overall block diagram of this approach is depicted in **Figure 6**.

5 TEXTURE PROPOSALS V1

The initial version of the V4Design TP module of is based on an efficient approach[4] with effective results, as we report in section 6.3 . The image transformation network is a deep residual convolutional neural network parameterised by weights W, and it transforms input images x into output images y via the mapping $y = f_W(x)$. Each loss function computes a scalar value measuring the difference between the output image and a target image. The image transformation network is trained using stochastic gradient descent to minimise a weighted combination of loss functions.

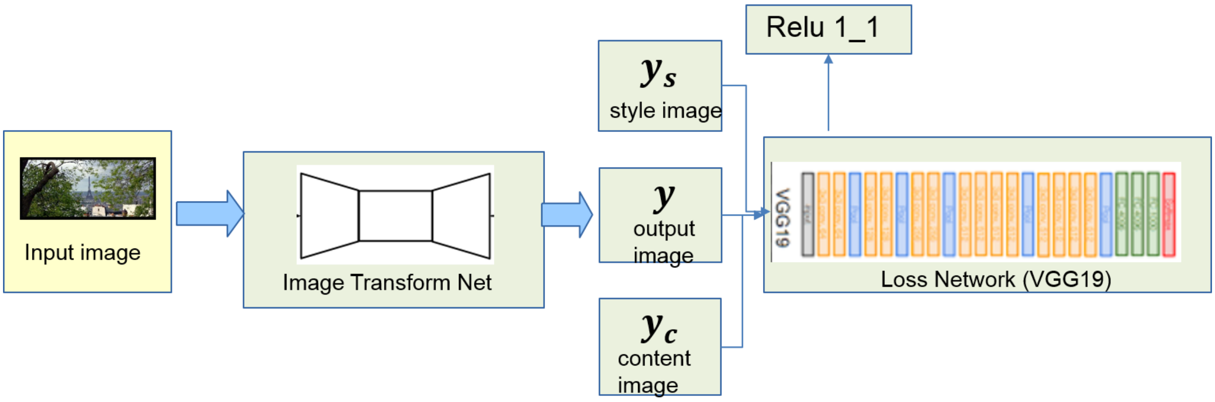


Figure 7: Texture proposals module framework

The image transformation network consists of five residual blocks and there are no pooling layers. Instead strided and fractionally strided convolutions are used for in-network down-sampling and up-sampling. All non-residual convolutional layers are followed by Ulyanov's instance normalisation (Ulyanov, 2017) and ReLU nonlinearities with the exception of the output layer, which instead uses a scaled $tanh$ to ensure the output image has pixels in the range [0,255]. The loss function of this implementation is close to the one described in (Gatys L. A., 2016), using VGG19 instead of VGG16 and shallower layers such as relu1_1.

For texture proposals the input and output are both colour images of shape 3×256×256 and the networks use two stride-2 convolutions to down-sample the input followed by several residual blocks and then two convolutional layers with stride 1/2 to up-sample. Although the input and output have the same size, there are several benefits to networks that down-sample and then up-sample. The framework of the implemented approach is depicted in the **Figure 7**.

[4] https://github.com/lengstrom/fast-style-transfer

6 EVALUATION

6.1 Aesthetics extractions in paintings

6.1.1 Dataset description

Initially, we did some experiments in a small dataset (Pandora) so as to verify our speculations and perform some parameter selection. And then we check our algorithm's generalisation capability in the more challenging and bigger WikiArt.

The Pandora paintings dataset has a collection of 18,720 paintings from many different resources. The image collection has been distributed among 18 style classes, having approximately 1,000 images. Engineers have ensured that only the relevant part of the images is shown and art experts also ensure that the artistic annotation is valid.

The Wikiart paintings dataset is an image collection of 81,472 paintings images, from more than 1,000 artists. This dataset contains 27 different styles and 45 different genres. Based on our knowledge, it is currently the largest digital art dataset publicly available for research purposes. 81,446 paintings are used for style classification, while only 10 genres with more than 1,500 paintings are chosen for genre classification, with a total of around 64,995 samples. Similarly, only a subset of 23 artists with more than 500 paintings is chosen, with total amount of 19,051 images for artist classification.

The Pandora paintings dataset has a collection of 18,720 paintings from many different resources. The image collection has been distributed among 18 style classes, having approximately 1,000 images. Engineers have ensured that only the relevant part of the images is shown and art experts also ensure that the artistic annotation is valid.

6.1.2 Settings

The input image size in the considered datasets is $n_h = 224$, $n_w = 224$ and $n_c = 3$ while the filter size which we use is $f = 3$ and for the max-pooling layers the pixel window is $w_p = 2$ and the stride is $s = 2$. The first two of the FC layers have 4096 channels and the third one has as channels as the number of the classes of each category. The first two of the FC layers have 4096 channels and the third one has as channels as the number of the classes of each category. More specifically, for style, genre and artist classification task in the Wikiart dataset the last channel has 27, 10 and 23 (classes) neurons respectively and for style classification task in Pandora dataset it has 18 neurons.

Our models are trained using the stochastic gradient descent (SGD) without weight decay and momentum, with a batch size of 64 examples for the Wikiart dataset and 32 examples for the Pandora dataset. The learning rate which we used is 0.003. The number of epochs varies for different tasks and number of training set. More specifically, for the Wikiart dataset we trained our model for 20 epochs in both artist and genre classification tasks and for 300 epochs for style classification task. Regarding the style classification task in the Pandora dataset, we trained our model for 136 epochs. We keep the original split for the Wikiart dataset as it is provided by authors and we use 4-fold cross-validation for the Pandora dataset, similar to the original paper. For our experiments we use Tensorflow[5]

[5] https://www.tensorflow.org/

backend and keras[6] deep learning neural network library in Python. We downloaded the weights from Places2 dataset for VGG16 architecture from Github[7]. We run our experiments on the GPU of NVIDIA FeForce GTX 1080 Ti.

The baseline methods that we use in the evaluation in the Wikiart dataset **Table 3** have been described in (Tan, 2016)**.** CNN refers to a deep model that was trained from scratch for each of the classification tasks. CNN-nofine, CNN-SVM, CNN-1000, and CNN-finetune are pre-trained model based on the ImageNet dataset. CNN-nofine is a CNN model without the fine-tuning process, while CNN-finetune is a model that has been fine-tuned. CNN-SVM replaces the last layer with a SVM classifier instead of the softmax layer.

The baseline methods that we use in the evaluation in the Pandora dataset **Table 4** have been described in (Florea, 2017). In particular, LeNet stands for the neural network presented in (LeCun, 1998), AlexNet stands for the neural network architecture originally introduced in (Krizhevsky, 2012), NiN refers to the ``Network in network'' representation presented in Lin et al., (2013) and finally, ResNet is the residual network architecture presented in (He, 2016). In **Table 4** size refers to the width and height of the input images, Layers to the number of layers, Rand refers to the case when initialisation was from scratch and FT−N refers to a pre-trained ImageNet instance with only the top N layers being re-trained by (Florea, 2017).

6.1.3 Results

In our results we show both qualitative and quantitative evaluation for our experiments in artwork classification tasks in the Wikiart and Pandora datasets.

Model	Style	Genre	Artist	Overall
CNN	42.96	65.45	54.39	54.27
CNN-nofine	45.95	69.24	67.02	60.74
CNN-SVM	45.95	69.24	67.02	60.17
CNN-1000	43.56	68.38	64.55	58.83
CNN-finetune	54.50	74.14	**76.11**	68.25
CNN-fc6	51.50	72.11	74.26	65.96
CNN-1024	53.38	73.75	76.02	67.72
Saleh and Elgammal	45.97	60.28	63.06	56.44
VGG16 pre-trained (Ours)	**55.70**	**75.10**	75.85	**68.78**

Table 3: Results for the Wikiart dataset evaluation

[6] https://keras.io/

[7] https://github.com/GKalliatakis/Keras-VGG16-places365/blob/master/vgg16_places_365.py

We observe in **Table 3** that in style classification task our approach in the Wikiart dataset has a relative increase of 2.15% and in genre classification task by 5.27%. Although in artist classification task the relative decrease is 0.34%, the overall performance is larger than the state-of-the-art baseline method. Artists may change their style throughout time and the classification is very difficult, such as the famous artist Pablo Picasso who has created paintings that belong to expressionism and other paintings that belong to cubism.

Model	Size	Layers	Time	RR
LeNet-Rand	32	14	$< 1h$	22.3
LeNet-Rand	64	16	$< 1h$	25.1
NiN-Rand	64	14	$< 1h$	26.5
AlexNet-Rand	224	8	$< 1h$	39.5
AlexNet-FT-3	224	8	$2h$	39.5
AlexNet-FT-4	224	8	$60h$	$\mathbf{56.5}$
ResNet-34-Rand	224	34	$2h$	47.8
VGG16 pre-trained (Ours)	$\mathbf{224}$	$\mathbf{16}$	$\mathbf{3.5h}$	56.11

Table 4: Results for the Pandora dataset evaluation

In **Table 4** our model shows comparable performance to the best performing baseline approach (AlexNet-FT-4) with the same image size in the Pandora dataset but in our training phase the duration was only 6% of the time needed to achieve the similar recognition rate (RR). In a comparable to us duration of the training phase (AlexNet-FT-3, ResNet-34-Rand) we observe that our method significantly outperforms all the other considered methods. That shows that using a pre-trained model can help get better results than training a new one without landscape characteristics.

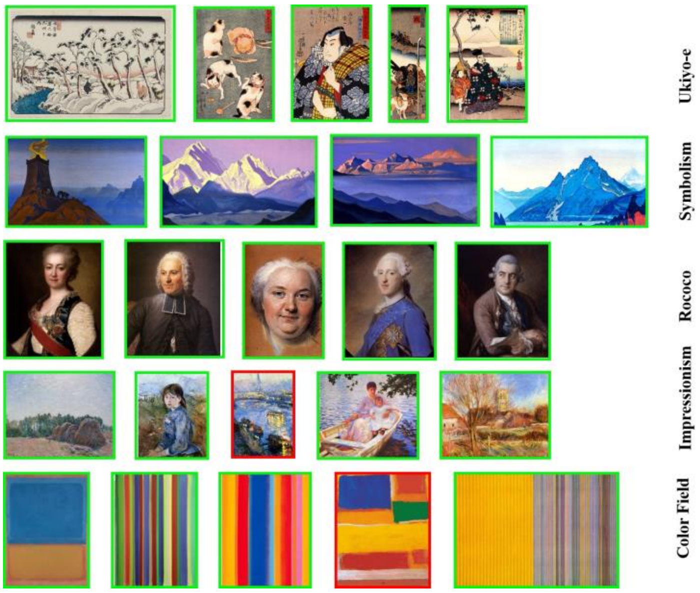


Figure 8: The top-5 predicted painting images based on their style of Wikiart dataset.

In **Figure 8** and **Figure 9** we present some qualitative results from both datasets in style classification task. We keep the top-ranked paintings images for 5 indicative classes. In the case of **Figure 8** the styles Impressionism and Colour-field of the Wikiart dataset include a falsely predicted image, but as can be seen from a non-expert in arts, both examples cannot be easily distinguished from the other correctly predicted paintings images. In **Figure 9** as we can observe there are false predictions for Abstract Art style and Cubism style, but similar to the **Figure 8**, it is difficult to perceive the differences.

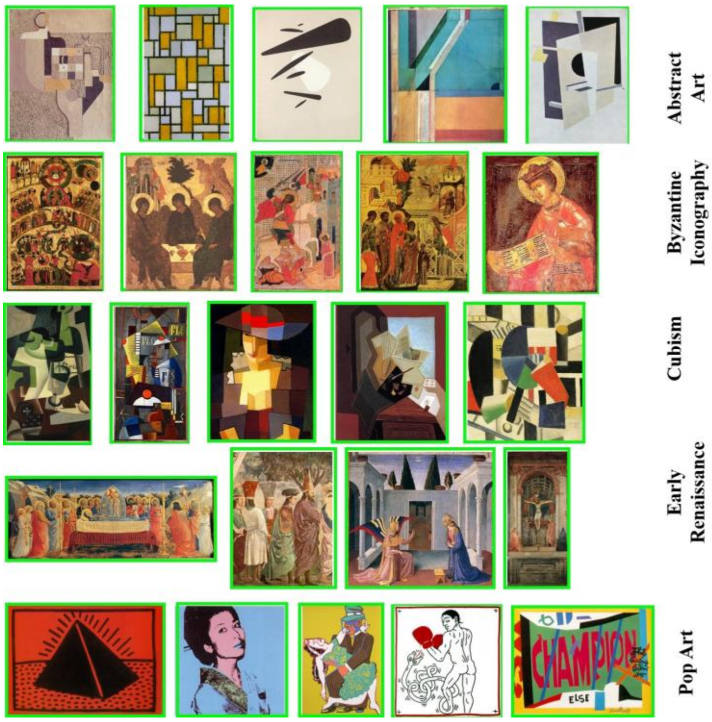
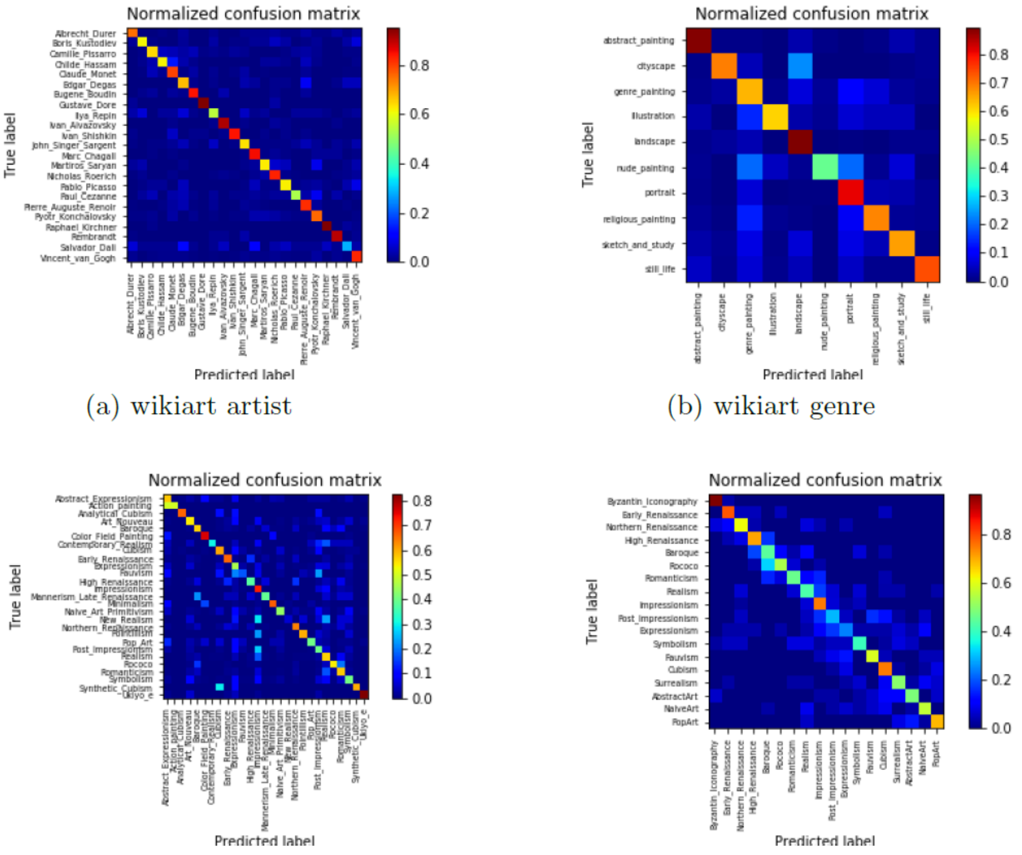


Figure 9: The top-5 predicted painting images based on their style of Pandora dataset.

The confusion matrices for all of our results for each dataset are presented in **Figure 10**, where the diagonal has clearly warm colour of the heat-map. We observe that some pairs of styles are difficult to be distinguished, such as cubism and synthetic-cubism, impressionism and post-impressionism and realism and new-realism.



(a) wikiart artist

(b) wikiart genre

(c) wikiart style

(d) pandora style

Figure 10: The normalised confusion matrices of classification by artist, genre and style on the Wikiart and Pandora datasets

6.2 Aesthetics extraction in movies

We have submitted 4 runs for valence/arousal prediction and their results are introduced in Table 5. In the experiments two evaluation measures are used: (a) Mean Square Error (MSE) and (b) Pearson Correlation Coefficient (r).

	Valence		Arousal		Fear
Run	**MSE**	**r**	**MSE**	**r**	**IoU**
1	396901706.564	0.079	1678218552.19	0.054	0.075
2	0.139	0.010	0.181	−0.022	0.065
3	**0.117**	**0.098**	**0.138**	*nan*	0.053
4	0.142	0.067	0.187	−0.029	0.063

Table 5: CERTH-ITI Predictions

We observe that the NN approach that we describe in the previous section has the best performance amongst all the others. Furthermore, it is worth mentioning that the linear regression model produces some extremely high scores, probably because the original feature vectors weren't neither discriminative nor adequate enough to create the regression model. However, PCA projection to lower dimensional space, with higher discriminative power show to solve this problem as it reduces the redundant noise and keep the most important features. Moreover, there is a "NaN" score for the Pearson measure in the arousal prediction scores, because we accidentally set the training value stable and so our model predicts the same score for all frames, but this score does not characterise our model, since it does not appear in any other prediction within the valence/arousal prediction sub-task.

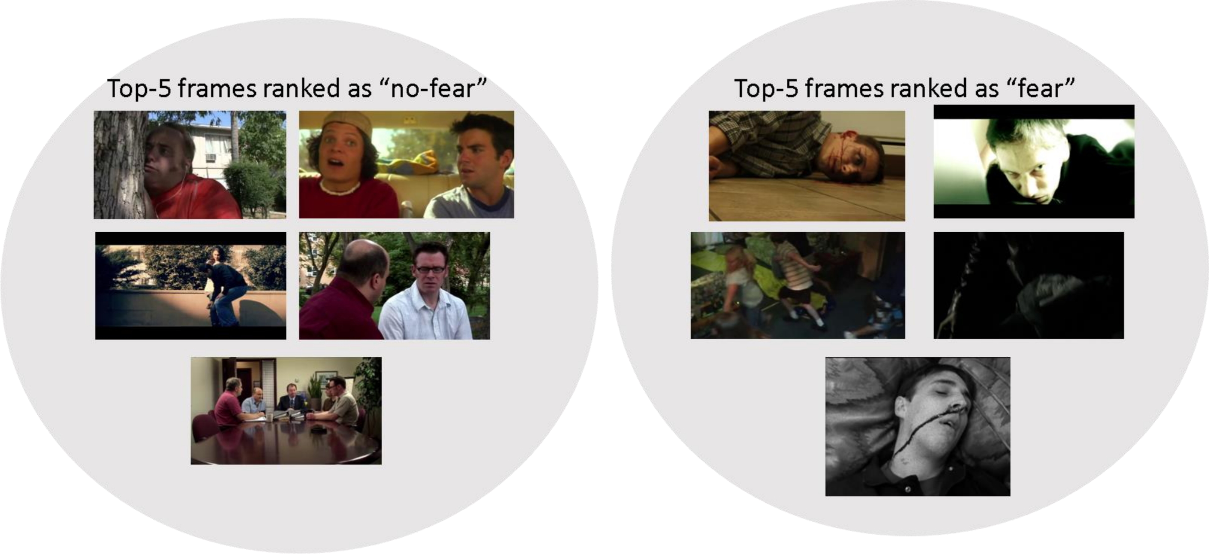

Top-5 frames ranked as “no-fear”
Top-5 frames ranked as “fear”

Figure 11: The Top-5 video frames ranked as “no-fear” and “fear” respectively.

We have also submitted 4 runs for fear prediction subtask and their results are also presented in Table 5 and are evaluated in terms of Intersection over Union (IoU).

From **Table 5** we see that, the best performance for the fear recognition subtask is Run 1, using all predicted scores of the pre-trained VGG16 model. In **Figure 11** we can observe the top-5 ranked video frames predicted as “no-fear” and “fear” respectively. In addition, our intuition to remove isolated predicted frames (Run 2), as they are not associated with any duration, did not perform better than Run 1, hence we miss significant information (video frames that invoke fear).

From **Table 7**: Comparison of all submitted results for arousal prediction.**Table 7**, **Table 8** and **Table 9** (in the Appendix A) we see the comparison of all submitted results from all participants for arousal prediction, valence prediction and fear prediction, respectively. As we observe from **Table 7**, CERTH-ITI results for MSE ranked in the top 5. From **Table 8** we can see that our runs ranked in top-15 and are comparable to the top-ranked results. From **Table 9** we observe that the first of our run has the best performance.

6.3 Texture proposals

The TP module has been evaluated with respect to processing time and qualitative results are also illustrated. We train one image transformation network per style target and on the Microsoft COCO dataset. In Table 6 we present the runtime of the implemented method for V4Design purposes and the baseline. Across different size of images, we see that runtime of the implemented method is approximately higher of the baseline’s method.

Image size	(Gatys L. A., 2016)	Implemented approach

Image size	(Gatys L. A., 2016)	Implemented approach
$256 * 256$	$15.86s$	$0.048s$
$3503 * 1609$	$> 214.44s$	$18.17s$

Table 6: Speed in seconds for texture proposals implementation vs Gatys.

In **Figure 12** and **Figure 13** we present a comparison of the baseline approach and the implemented one. The given content image is presented on the left and the style image on the top. The middle column presents Gatys results and the right the results of the implemented approach. We observe that in **Figure 12** and **Figure 13** the examples of the implemented method and the baseline are comparable in this qualitative way. On the one hand we see that in **Figure 12** the faces fade out in the case of the baseline approach, while in **Figure 13** the baby's face becomes more blurry in the implemented significantly faster method.

Finally, **Figure 14** and **Figure 15** depict how texture proposals can transfer the paintings' style to architectural buildings, so that it can be become clearer how target images can change and provide the appropriate input to 3D reconstruction (T4.3) and re-texturize the 3D models appropriately. The main goal of our algorithm here was to transfer the background and foreground features from the paintings to the appropriate regions in the target architecture images. This strategy will retain the aesthetics of the source image and will give the emotion that it could have been created from the same person/painter under the scope of the same movement (i.e. surrealism, cubism, etc)

Due the fact that in V4Design we want to transfer textures not only in images but also in videos, we choose to implement an algorithm which could produce results with almost the same quality but in significantly less time.

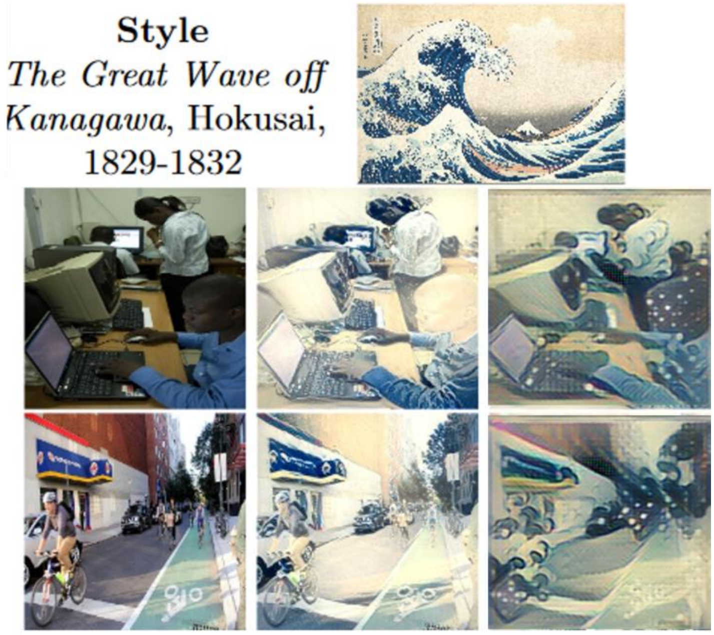


Figure 12: Qualitative comparison of texture proposals implementation vs Gatys et al. using as style the painting "The great wave off Kanagawa".

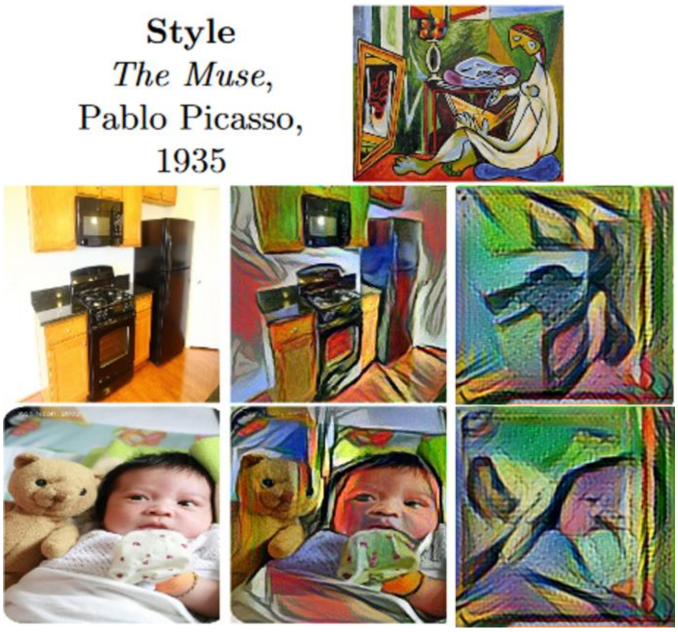
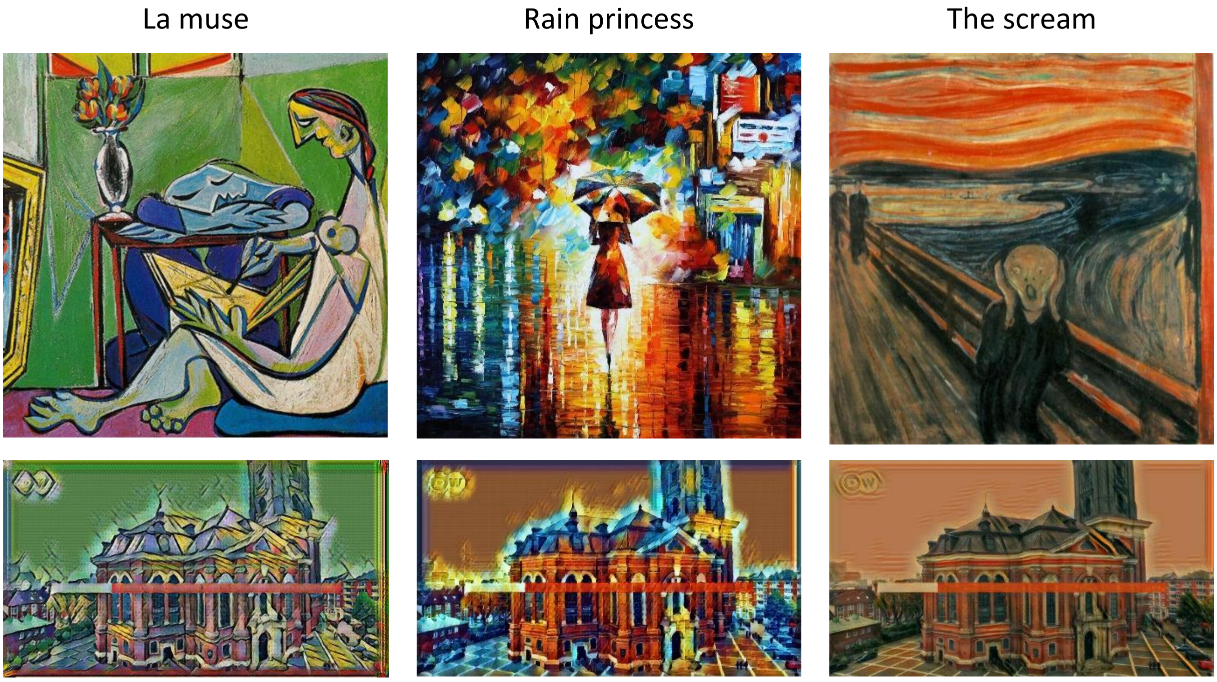


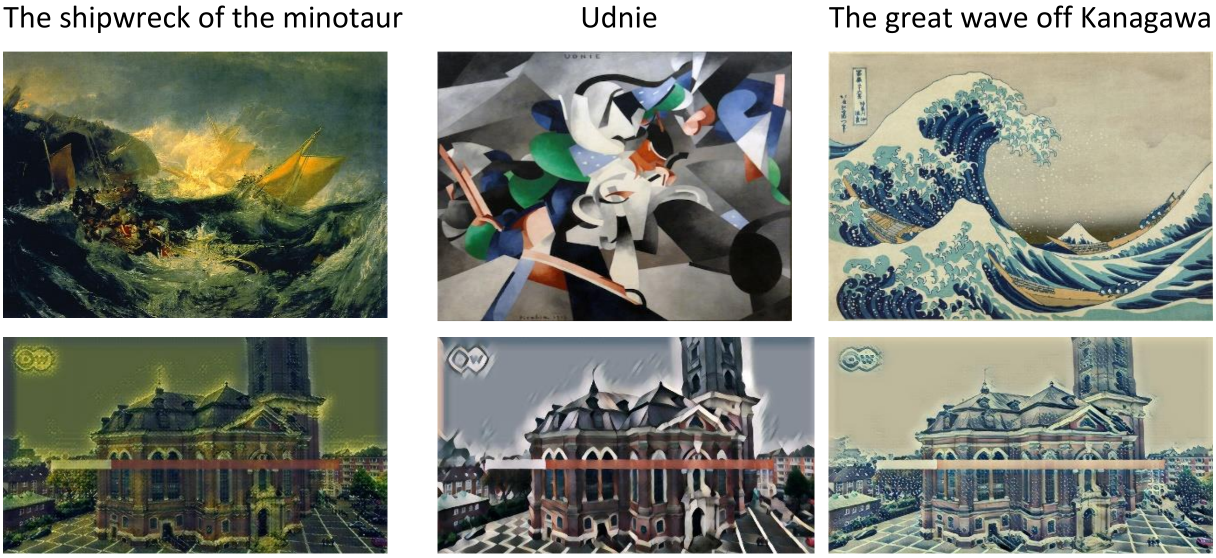
Figure 13: Qualitative comparison of texture proposals implementation vs Gatys et al., 2016 using as style the painting “The Muse”.


La muse

Rain princess

The scream

Figure 14. Texture proposals from famous paintings to architecture building (v1)



The shipwreck of the minotaur

Udnie

The great wave off Kanagawa

Figure 15. Texture proposals from famous paintings to architecture building (v2)

7 CONCLUSIONS AND NEXT STEPS

7.1 Conclusions

In aesthetics extraction of paintings, we examined the performance of transfer learning using the VGG16 architecture pre-trained on benchmark Places2 dataset in paintings classification problem to extract style, genre and artist characteristics per painting image. We applied our approach on two publicly available paintings images datasets. Our method outperforms state-of-the-art approaches in terms of effectiveness and efficiency. In aesthetics extraction, such as style classification, models trained from scratch each time demand a lot of computational resources and domain-specific annotated datasets. Our proposed method using transfer learning, addresses this issue by exploiting the knowledge gained by the Places2 external collection, while passing features from places images to style classification of paintings images. To that end, Convolutional Neural Networks like VGG16 which are pre-trained on large scale datasets, such as Places2, can be used as fixed feature extractor improving the efficiency of existing state-of-the-art models in aesthetics extraction problem and offers more effective solutions.

In aesthetics extraction of movies, the results in valence/arousal prediction subtask shows that according to MSE, our best result obtained in one of our runs (Run 3) for both valence and arousal, while regarding to Pearson Correlation Coefficient, another run (Run 1) had the best performance for arousal and the second best performance for valence estimations, respectively. The Pearson correlation is able to measure linear correlations between two or more variables. However, the MSE is obtained by a sum of squared deviations between predicted and ground-truth values, no matter if they are linearly correlated or not. The results of the fear prediction subtask show that the inclusion of audio features failed to enhance the classification performance. This could be due to several reasons, with the prominent one to be the incapability of performing data augmentation on audio features such as in the case of visual analysis.

In texture proposals, we have combined the benefits of feed-forward image transformation and optimisation-based methods for image generation by training feed-forward transformation networks. We have applied this method to style transfer where we achieve comparable performance and drastically improved speed compared to baseline method.

7.2 Next steps

7.2.1 Aesthetics extraction

In the future, we plan to examine the aesthetics extraction method to other datasets of different nature that will not only include paintings but any kind of artworks, and also in architecture so as to define how old a building is based on its structural elements, architecture style and creator of a monument.

In aesthetics extraction of movies, we plan to overcome the drawbacks by using classification methods able to handle unbalanced training sets, such as penalized models, or by enriching the training set with external annotated datasets and by exploring more efficient fusion methods, such as performing classification on fused audio-visual features, instead of a posterior combining separate classification results.

7.2.2 Texture proposals

We plan to propose a novel framework which will be able to generate a new image that is a combination of content image and more than one styles. In this way, new textures will be proposed and transferred to a whole image. Moreover, the developed Deep learning framework for texture proposals will be transferred only to a specific part of image or video that includes a target object or building. Target objects and buildings will be localised in a spatio-temporal way in multimedia content using a method which provides a segmentation mask per object in each image or video frame. In this way, it will be possible to transfer style from background to background or foreground to foreground respectively.

8 REFERENCES

Baveye, Y. D. (2015). Deep learning vs. kernel methods: Performance for emotion prediction in videos. *International Conference on Affective Computing and Intelligent Interaction (ACII)* (pp. 77-83). IEEE.

Elgammal, A. &. (2015). Quantifying Creativity in Art Networks. *Sixth International Conference on Computational Creativity*, (p. 39).

Elgammal, A. L. (2017). Can: Creative adversarial networks, generating" art" by learning about styles and deviating from style norms.

Eyben, F. W. (2013). Recent developments in opensmile, the munich open-source multimedia feature extractor. *21st ACM international conference on Multimedia* (pp. 835-838). ACM.

Florea, C. T. (2017). Artistic movement recognition by boosted fusion of color structure and topographic description. *Winter Conference on Applications of Computer Vision (WACV)* (pp. 569-577). IEEE.

Gatys, L. A. (2016). Image style transfer using convolutional neural networks. *computer vision and pattern recognition* (pp. 2414-2423). IEEE.

Gatys, L. A. (2016). Image style transfer using convolutional neural networks. *computer vision and pattern recognition.* IEEE.

Guo, X. Z. (2018). Group-Level Emotion Recognition using Hybrid Deep Models based on Faces, Scenes, Skeletons and Visual Attentions. *International Conference on Multimodal Interaction* (pp. 635-639). ACM.

He, K. Z. (2016). Deep residual learning for image recognition. *computer vision and pattern recognition* (pp. 770-778). IEEE.

Huang, X. &. (2017). Arbitrary style transfer in real-time with adaptive instance normalization. *Conference on Computer Vision* (pp. 1501-1510). IEEE.

Jain, D. K. (2019). Extended deep neural network for facial emotion recognition. *Pattern Recognition Letters*, 69-74.

Karayev, S. T. (2013). Recognizing image style.

Krizhevsky, A. S. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, (pp. 1097-1105).

LeCun, Y. B. (1998). Gradient-based learning applied to document recognition. . *Proceedings of the IEEE*, (pp. 2278-2324).

Li, C. &. (2009). Aesthetic visual quality assessment of paintings. *Journal of selected topics in Signal Processing* (pp. 236-252). IEEE.

Li, S. D. (2017). Reliable crowdsourcing and deep locality-preserving learning for expression recognition in the wild. . *Computer Vision and Pattern Recognition* (pp. 2852-2861). IEEE.

Lu, X. L. (2014). Rapid: Rating pictorial aesthetics using deep learning. *ACM international conference on Multimedia* (pp. 457-466). ACM.

Lu, X. L. (2015). Deep multi-patch aggregation network for image style, aesthetics, and quality estimation. *International Conference on Computer Vision* (pp. 990-998). IEEE.

Murray, N. M. (2012). Murray, N., Marchesotti, L., & Perronnin, F. (2012, June). AVA: A large-scale database for aesthetic visual analysis. *Computer Vision and Pattern Recognition* (pp. 2408-2415). IEEE.

Nishiyama, M. O. (2011). Aesthetic quality classification of photographs based on color harmony. *CVPR* (pp. 33-40). IEEE.

Oliva, A. (2005). Gist of the scene. *Neurobiology of attention* (pp. 251-256). Academic Press.

Saleh, B. &. (2015). Large-scale classification of fine-art paintings: Learning the right metric on the right feature. .

Shalunts, G. H. (2011). Architectural style classification of building facade windows. . *International Symposium on Visual Computing* (pp. 280-289). Berlin: Springer.

Shamir, L. M. (2010). Impressionism, expressionism, surrealism: Automated recognition of painters and schools of art. . *Transactions on Applied Perception .* ACM.

Sheng, L. L. (2018). Avatar-net: Multi-scale zero-shot style transfer by feature decoration. *Computer Vision and Pattern Recognition* (pp. 8242-8250). IEEE.

Simonyan, K. &. (2014). Very deep convolutional networks for large-scale image recognition.

Tan, W. R. (2016). Ceci n'est pas une pipe: A deep convolutional network for fine-art paintings classification. . *International Conference on Image Processing (ICIP)* (pp. 3703-3707). IEEE.

Ulyanov, D. V. (2017). Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. *Computer Vision and Pattern Recognition* (pp. 6924-6932). IEEE.

Xu, Z. W. (2018). Beyond textures: Learning from multi-domain artistic images for arbitrary style transfer.

Zeppelzauer, M. D. (2018). Automatic Prediction of Building Age from Photographs. *International Conference on Multimedia Retrieval* (pp. 126-134). ACM.

Zhang, S. Z. (2018). Learning affective features with a hybrid deep model for audio–visual emotion recognition. *Circuits and Systems for Video Technology* (pp. 3030-3043). IEEE Transactions.

Zhao, S. G. (2014). Exploring principles-of-art features for image emotion recognition. *ACM international conference on Multimedia* (pp. 47-56). ACM.

Zhou, B. L. (2014). Learning deep features for scene recognition using places database. . *Advances in neural information processing systems* , (pp. 487-495).

Zhou, B. L. (2018). Places: A 10 million image database for scene recognition. *pattern analysis and machine intelligence* (pp. 1452-1464). IEEE transactions.

A Appendix

A.1. AE from movies – results from MediaEval 2018, Emotional impact of movies task

A.1.1 Arousal

Run	MSE	Pearson's CC
me18ei_CERTH_ITI_valence_arousal_3.txt	**0.138286415202**	nan
me18ei_GOOGLE_valence_arousal_4.txt	0.139560166014	0.351327564809
me18ei_GOOGLE_valence_arousal_5.txt	0.133369539381	0.335834232398
me18ei_GOOGLE_valence_arousal_1.txt	0.138411846123	0.291088878635
me18ei_GOOGLE_valence_arousal_3.txt	0.177832306641	0.277254974466
me18ei_GOOGLE_valence_arousal_2.txt	0.147901400305	0.195712338604
me18ei_MIC-TJU_valence_arousal_4.txt	0.136243598085	0.174860820616
me18ei_MIC-TJU_valence_arousal_2.txt	0.135979522567	0.15545986132
me18ei_MIC-TJU_valence_arousal_5.txt	0.139498519082	0.152258587102
me18ei_MIC-TJU_valence_arousal_3.txt	0.140558553627	0.143101610798
me18ei_MIC-TJU_valence_arousal_1.txt	0.146340827619	0.1115705846904
me18ei_HCMUS_valence_arousal_5.txt	0.172824182606	0.0912361435628
me18ei_THU-HCSI_valence_arousal_1.txt	0.14136851448	0.0870179936725
me18ei_HKBU_valence_arousal_1.txt	0.149300305213	0.0828098770876
me18ei_THU-HCSI_valence_arousal_3.txt	0.139937785313	0.0760704473072
me18ei_HCMUS_valence_arousal_2.txt	0.170550861649	0.075251349679
me18ei_HCMUS_valence_arousal_3.txt	0.174435750215	0.069782947227
me18ei_HKBU_valence_arousal_2.txt	0.157470101131	0.0650681336943
me18ei_THU-HCSI_valence_arousal_4.txt	0.139613465922	0.0612147053487
me18ei_CERTH-ITI_valence_arousal_1.txt	1678218552.19	0.0540306063467
me18ei_HCMUS_valence_arousal_1.txt	0.174488017375	0.0528274725998
me18ei_HKBU_valence_arousal_3.txt	0.160887855582	0.0487654244123
me18ei_HKBU_valence_arousal_4.txt	0.162365707296	0.0255841574264
me18ei_HCMUS_valence_arousal_4.txt	0.17901120145	0.0187704038881
me18ei_MVIP-CSIR_valence_arousal_2.txt	0.192105171407	0.015621155911
me18ei_THU-HCSI_valence_arousal_5.txt	0.146026279788	−0.0687220073305

Run	MSE	Pearson's CC
me18ei_CERTH-ITI_valence_arousal_4.txt	0.187582486852	−0.0298418288021
me18ei_CERTH-ITI_valence_arousal_2.txt	0.180882313892	−0.0228631989841
me18ei_THU-HCSI_valence_arousal_2.txt	0.139895260551	−0.0180885184353
me18ei_MVIP-CSIR_valence_arousal_1.txt	0.16649382582	−0.014709350579

Table 7: Comparison of all submitted results for arousal prediction.

A.1.2 Valence

Run	MSE	Pearson's CC
me18ei_THU-HCSI_valence_arousal_3.txt	0.0923511980506	0.304751470641
me18ei_MIC-TJU_valence_arousal_2.txt	0.0903759843852	0.300844590286
me18ei_GOOGLE_valence_arousal_4.txt	0.107254414605	0.277898745944
me18ei_MIC-TJU_valence_arousal_1.txt	0.091418198244	0.27518324591
me18ei_MIC-TJU_valence_arousal_3.txt	0.0916310713828	0.263262794717
me18ei_MIC-TJU_valence_arousal_4.txt	0.091051765983	0.256683852944
me18ei_THU-HCSI_valence_arousal_5.txt	0.0944106509426	0.251071756996
me18ei_MIC_TJU_valence_arousal_4.txt	0.0979527032913	0.24220302749
me18ei_GOOGLE_valence_arousal_3.txt	0.113349921807	0.188310930424
me18ei_THU-HCSI_valence_arousal_2.txt	0.103595999918	0.188310930424
me18ei_GOOGLE_valence_arousal_5.txt	0.0836665990219	0.178593664961
me18ei_THU-HCSI_valence_arousal_1.txt	0.102139190209	0.171404475347
me18ei_HCMUS_valence_arousal_2.txt	0.115047011932	0.145658087768
me18ei_HCMUS_valence_arousal_3.txt	0.119436755421	0.145136006527
me18ei_HCMUS_valence_arousal_5.txt	0.115264011093	0.143068232802
me18ei_HCMUS_valence_arousal_4.txt	0.117319308413	0.140973336023
me18ei_GOOGLE_valence_arousal_2.txt	0.0945491451519	0.137593837672
me18ei_GOOGLE_valence_arousal_1.txt	0.119256148823	0.117494557359
me18ei_HKBU_valence_arousal_4.txt	0.107676846396	0.114261840951
me18ei_HCMUS_valence_arousal_1.txt	0.119363701445	0.106655704211
me18ei_CERTH_ITI_valence_arousal_3.t	**0.117381489285**	0.0989509705722

Run	MSE	Pearson’s CC
xt		
me18ei_HKBU_valence_arousal_3.txt	0.108938257945	0.0872360857036
me18ei_CERTH-ITI_valence_arousal_1.txt	396301706.564	0.07937973778
me18ei_CERTH-ITI_valence_arousal_4.txt	0.142217167106	0.0674070168499
me18ei_HKBU_valence_arousal_1.txt	0.101633469474	0.0049975600848
me18ei_MVIP-CSIR_valence_arousal_1.txt	0.135800414076	0.0483505345935
me18ei_HKBU_valence_arousal_2.txt	0.108977520373	0.00164717741036
me18ei_CERTH-ITI_valence_arousal_2.txt	0.139936458773	0.010513382559
me18ei_MVIP-CSIR_valence_arousal_2.txt	0.172886531165	−0.027629113425

Table 8: Comparison of all submitted results for valence prediction.

A.1.3 **Fear**

Run	IoU
me18ei_MIC-TJU_fear_4.txt	0.15750375111
me18ei_MIC-TJU_fear_5.txt	0.149688294723
me18ei_MIC-TJU_fear_1.txt	0.143597900392
me18ei_MIC-TJU_fear_3.txt	0.13668734977
me18ei_MIC-TJU_fear_2.txt	0.129003228743
me18ei_IM-JAIC_fear_4.txt	0.1119923843351
me18ei_HKBU_fear_1.txt	0.105277877183
me18ei_IM-JAIC_fear_5.txt	0.0987356534746
me18ei_IM-JAIC_fear_3.txt	0.0874229561062
me18ei_CERTH-ITI_fear_1.txt	**0.0758513724696**
me18ei_ IM-JAIC_fear_2.txt	0.0750745975147
me18ei_ CERTH-ITI_fear_2.txt	0.0659517305098
me18ei_ IM-JAIC_fear _1.txt	0.0649595100541
me18ei_ CERTH-ITI_fear_4.txt	0.0637160516567
me18ei_HKBU_fear_2.txt	0.0612654401029

Run	IoU
me18ei_ CERTH-ITI_fear_3.txt	0.0535523327493
me18ei_ HKBU_fear_3.txt	0.0360481326149
me18ei_ HKBU_fear_4.txt	0.019605622133

Table 9: Comparison of all submitted results for fear prediction.