INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA DE INTELIGÊNCIA ARTIFICIAL




Remoção de Objetos em Imagens do Mercado Imobiliário usando Inpainting

ANDRÉ FERREIRA ALVES DOS REIS
Outubro de 2023

Real estate application of image object removal inpainting techniques

André Ferreira Alves dos Reis
Student No.: 1181207

Dissertation for obtaining the Master's Degree in Artificial Intelligence Engineering

Supervisor: Doutor Carlos Fernando da Silva Ramos, Professor Coordenador Principal do Instituto Superior de Engenharia do Instituto Politécnico do Porto

Evaluation Committee:
President:
Doutor António Constantino Lopes Martins, Professor Adjunto do Instituto Superior de Engenharia do Instituto Politécnico do Porto

Members:
Doutor Carlos Fernando da Silva Ramos, Professor Coordenador Principal do Instituto Superior de Engenharia do Instituto Politécnico do Porto

Doutor Luís Filipe Oliveira Gomes, Investigador Auxiliar do Instituto Superior de Engenharia do Instituto Politécnico do Porto

Porto, October 2023

"A kilogramme of steel is heavier than a kilogramme of feathers!"

Brian "Limmy" Limond

Abstract

The real estate market is one of the most valuable and influential markets in the world. An essential part of this market is the first contact with potential customers through property listings on real estate websites. To make listings more attractive, professionals often hire external services or use image editing software to improve the quality of the images shown. Real estate agents often use tools to remove unwanted objects from images to declutter rooms, draw attention to the property, and eliminate uncontrollable environmental elements. However, existing solutions are not cost-effective for real estate agencies, as they have to pay for each image and the results can take up to two days to be delivered.

In recent years, the use of deep learning artificial intelligence algorithms has revolutionised image removal technology, with current technologies capable of producing natural and realistic results. This paper focusses on the development of a deep learning inpainting image object removal solution for real estate images to be integrated into Maxwork, the back-office portal used by RE/MAX Portugal, one of the largest real estate companies in Portugal.

This solution can remove objects from real estate images and contains several additional features, including the ability to undo and redo, and compare the original image with the most recent result. It uses the LaMa inpainting deep learning model, which proved to be more effective than other state-of-the-art models. The effectiveness of the solution was evaluated with several objective metrics, such as Fréchet Inception Distance (FID) and Learned Perceptual Image Patch Similarity (LPIPS), and subjectively with user feedback that gave an average rating of 4.31 for ease of use and 3.90 for satisfaction with the results obtained. Furthermore, ethical considerations related to image editing in the real estate sector are discussed to ensure a transparent and honest use of the solution.

This paper also outlines the details of an experiment to train a new deep learning inpainting model using a collection of around 100,000 real estate images from past sold listings. However, the results of the experiment showed that the trained model was not able to outperform the pre-trained LaMa model, as it scored worse across all metrics. Additionally, this paper provides research on existing real estate image object removal solutions, the current application fields of deep learning inpainting models, and the evolution of inpainting models from traditional methods to current state-of-the-art deep learning models.

Keywords: Real Estate, Inpainting, Image Object Removal, Computer Vision, LaMa

Resumo

O mercado imobiliário é um dos mercados mais valiosos e influentes do mundo. Uma parte essencial desse mercado é o primeiro contato com potenciais clientes por meio de anúncios de imóveis em sites imobiliários. Para tornar estes anúncios mais atrativos, os profissionais muitas vezes contratam serviços externos ou usam software de edição de imagens para melhorar a qualidade das imagens exibidas. Agentes imobiliários frequentemente usam ferramentas para remover objetos indesejados das imagens, a fim de desobstruir os quartos, chamar a atenção para a propriedade e eliminar elementos ambientais incontroláveis. No entanto, as soluções existentes não são economicamente viáveis para agências imobiliárias, pois elas têm de pagar por cada imagem e os resultados podem levar até dois dias para serem entregues.

Nos últimos anos, o uso de algoritmos de inteligência artificial de aprendizagem profunda revolucionou a tecnologia de remoção de imagens, com as tecnologias atuais capazes de produzir resultados naturais e realistas. Este trabalho foca-se no desenvolvimento de uma solução de remoção de objetos de imagem usando aprendizagem profunda para imagens imobiliárias, a ser integrada no Maxwork, o portal de *backoffice* usado pela RE/MAX Portugal, uma das maiores empresas imobiliárias em Portugal.

Esta solução é capaz de remover objetos de imagens imobiliárias e contém diversas funcionalidades adicionais, incluindo a capacidade de desfazer e refazer ações, bem como comparar a imagem original com o resultado mais recente. Ela utiliza o modelo de aprendizagem profunda LaMa, que se mostrou mais eficaz do que outros modelos estado da arte. A eficácia da solução foi avaliada por meio de várias métricas objetivas, como *Fréchet Inception Distance* (FID) e *Learned Perceptual Image Patch Similarity* (LPIPS), e subjetivamente com feedback de um grupo de utilizadores, que deram uma classificação média de 4.31 para a facilidade de uso e 3.90 para satisfação com os resultados obtidos. Além disso, são discutidas considerações éticas relacionadas à edição de imagens no setor imobiliário, para garantir um uso transparente e honesto da solução.

Este trabalho também descreve os detalhes de uma experiência para treinar um novo modelo de remoção de objetos de imagens usando uma coleção de cerca de 100.000 imagens imobiliárias de anúncios passados. No entanto, os resultados da experiência mostraram que o modelo treinado não conseguiu superar o modelo LaMa pré-treinado, pois obteve resultados piores em todas as métricas. Além disso, este trabalho oferece uma pesquisa sobre soluções existentes de remoção de objetos de imagens imobiliárias, e os campos de aplicação atuais e a evolução deste tipo de modelos, desde métodos tradicionais até modelos estado da arte de aprendizagem profunda.

Palavras-chave: Imobiliário, *Inpainting*, Remoção Objetos Imagem, Visão Computacional, LaMa

Acknowledgement

I am immensely grateful to my family, friends, colleagues, and teachers for their support.

I am grateful to my close friends João Campos, José Magalhães, Rúben Teixeira, Diogo Formosinho, and João Santos for their constant encouragement.

I also would like to thank my colleagues João Sousa, Luís Maia, Frederico Junqueira, and Luís Araújo for their invaluable input in the project.

Lastly, I would like to express my appreciation to Carlos Ramos, my supervisor, for his generous contribution of time and advice that made this project a success.

Contents

List of Figures

List of Tables

List of Source Code

List of Acronyms

API	Application Programming Interface.
AWS	Amazon Web Services.
CAM	Contextual Attention Module.
CE	Context Encoder.
CNN	Convolutional Neural Network.
CPU	Central Processing Unit.
CVF	Computer Vision Foundation.
CVPR	Computer Vision and Pattern Recognition.
FFC	Fast Fourier Convolution.
FFHQ	Flicker Faces High Quality.
FFM	Fast Marching Method.
FID	Fréchet Inception Distance.
GAN	Generative Adversarial Network.
GDPR	General Data Protection Regulation.
GPU	Graphics Processing Unit.
IEEE	Institute of Electrical and Electronics Engineers.
KNN	K-Nearest Neighbour.
LDM	Latent Diffusion Model.
LPIPS	Learned Perceptual Image Patch Similarity.
MAT	Mask-Aware Transformer.
NNF	Nearest Neighbour Field.
NTIRE	New Trends in Image Restoration and Enhancement.
P-IDS	Paired Inception Discriminative Score.
PDE	Partial Differential Equation.
PSNR	Peak Signal-to-Noise Ratio.
SN	Spectral Normalization.
SSIM	Structural Similarity Index Measure.
SVM	Support Vector Machine.

U-IDS Unpaired Inception Discriminative Score.
USD United States Dollar.

VAE Variational Autoencoder.

WGAN Wasserstein Generative Adversarial Network.

Chapter 1

Introduction

This chapter provides an overview of the project, including the issue being addressed, the objectives, and the contributions of the proposed solution.

1.1 Context

Image inpainting is the technique of restoring damaged or missing parts of an image. Initially, this term was used to refer to the repair of physical artwork, such as paintings and photographs, but it has since been extended to include digital media. Nowadays, inpainting is mainly divided into three applications: image restoration, object removal, and image generation. In recent years, the use of deep learning artificial intelligence algorithms has revolutionised image inpainting, with current technologies capable of producing natural and realistic results.

This thesis concentrates on the development of a deep learning image inpainting object removal solution for real estate images to be incorporated into Maxwork, a real estate back-office portal used by RE/MAX Portugal, one of the biggest real estate companies in Portugal, to manage all aspects of the sale and purchase of a property.

1.2 Problem description

An analysis report for 2022 [1] estimated that the global real estate market value in 2022 is 3.81 trillion United States Dollar (USD). It is projected that this value will reach 5.85 trillion USD by 2030, making it one of the most valuable and influential markets in the world.

One of the most important elements of this market is the first contact with potential customers through property listings on real estate websites. To draw more attention to a listing, professionals strive to make their listings more attractive by writing compelling descriptions and taking eye-catching photos. However, for various reasons, such as lack of time and inadequate equipment, it is not always possible to take perfect pictures. To address this issue, professionals often hire external services or use image editing software to enhance the quality of images by adjusting the composition, adjusting the lighting, and eliminating unwanted objects.

Real estate professionals can benefit from object removal tools to enhance the quality of their images. It can be used to remove furniture not included in the sale, personal items of the owner, wall imperfections that will be fixed before the sale, and construction tools in indoor images. This helps to declutter rooms and present the property in a "blank" state. For outdoor images, object removal can be used to eliminate uncontrollable environmental

elements such as parked cars, animals or people passing by, and trash bins. This reduces distractions for potential customers, since outdoor images are usually used as the listing thumbnail, and focusses attention on the property.

Currently, there are services such as Phixer [2] and Plan It All [3] that can be used to remove objects from real estate images. However, this can be inefficient for real estate agencies due to the amount of properties they handle daily, as it can take up to 48 hours to receive the results, consuming both time and money. Artificial intelligence image inpainting can be used to address this issue, as it can provide results in near real time. In recent years, inpainting models have been applied to various tasks, from restoring old photos (Wan et al. [4, 5]), removing text (Xie et al. [6]), and object removal (Bartl et al. [7]). An example of image object removal using inpainting is shown in Figure 1.1.

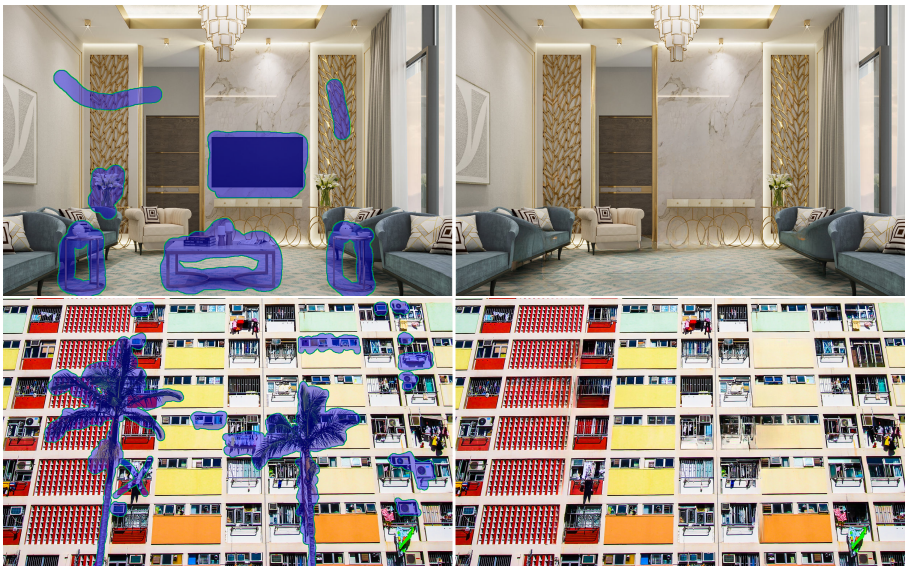

Figure 1.1: Examples of inpainting object removal using the LaMa model [8]. On the left is the original image with a selection mask applied and on the right is the resulting image after object removal.

1.3 Objectives and contributions

The primary objective of this project is the development of an artificial intelligence-based inpainting solution for the removal of real estate image objects that can be used by real estate professionals. Due to time constraints and considering the results of recent inpainting solutions, the focus was not on the construction of a new model from the ground up, but rather on adapting an existing open-source solution to the issue.

To accomplish this, it is necessary to perform a comprehensive review of the inpainting techniques developed over the years and the existing solutions for the elimination of objects from real estate images. Furthermore, this research contributes to the field of study by providing a summary of the current state of inpainting technology. The complete objectives and contributions of the project are described in the following list.

- Study on the current object removal solutions for real estate images;
- Study on the evolution of artificially intelligent inpainting solutions;
- Development of an artificially intelligent inpainting solution for real estate image object removal;
- Integration of the image object removal solution with the Maxwork portal;
- Evaluation of the performance and viability of the developed solution;
- Experimentation of an open-source inpainting solution adapted for real estate images;

1.4 Document structure

This document is structured into five chapters: Introduction; Literature review; Solution development; Experimentation and results; and Conclusions.

This chapter provides the context for the project, outlining the issue being addressed and the objectives and main contributions of the solution. The following chapter presents the results of a literature review on the main topics of the project. It begins by outlining the research methodology, followed by a comprehensive overview of the development of inpainting artificial intelligence algorithms, including past and current solutions. It then examines the current areas of application of inpainting algorithms and existing solutions for the removal of objects from real estate images, concluding with a summary and discussion of the findings of the literature review.

In the third chapter, a comprehensive overview of the solution development process is provided, including requirements analysis, architectural design, and the development of the various components. Additionally, it contains a section on the social implications of the solution, such as ethical considerations and data security. The fourth chapter concentrates on the experiment conducted to train the deep learning model with real estate images and the evaluation of this experiment in comparison to existing models. The last chapter provides a conclusion by contrasting the results achieved with the goals established at the beginning of the project and considering potential future enhancements.

Chapter 2

Literature review

This chapter presents the results of a literature review conducted to answer the following questions.

- How have inpainting artificial intelligence models evolved in the past years?
- What are the current state-of-the-art deep learning inpainting models?
- What are the current application fields of inpainting technologies?
- What are some of the existing solutions for image object removal?

The literature review was conducted with the objective of better understanding the state-of-the-art inpainting technologies. The research mainly used the Institute of Electrical and Electronics Engineers (IEEE) the and ScienceDirect databases, focussing on articles written in English up to 3 years ago (2020), which were relevant to the questions defined. The results of each research question are described in each section, with a final section discussing the results obtained.

2.1 The evolution of inpainting models

Throughout the years, the field of inpainting has seen the emergence of increasingly sophisticated artificial intelligence algorithms and techniques for filling in missing or damaged sections of an image. Initially, traditional inpainting methods relied on mathematical and machine learning algorithms, but the most significant progress in this area has been made with the introduction of deep learning models, such as Convolutional Neural Networks (CNNs) and Generative Adversarial Networks (GANs).

Several review papers have been published that provide an overview of the development of inpainting techniques. This section summarises the information from five of these review papers [9–13], highlighting some of the works that have been instrumental in the advancement of inpainting methods.

2.1.1 Traditional methods

The works of Efros and Leung [14] in 1999 and Bertalmio et al. [15] in 2000 are considered the pioneers of modern image inpainting techniques, defining two of the most popular traditional inpainting methods, patch-based and diffusion-based.

Efros and Leung [14] proposed a non-parametric sampling method to generate textures in images. This method divides an image into small patches and then selects the most similar patches to the neighbourhood around the damaged area to fill in the missing pixels,

repeating the process until the damaged area is filled. Barnes et al. [16] later improved on this method with the introduction of PatchMatch. This new model drastically reduced the memory and computational cost of the search process in the original model by using a rapid randomised Nearest Neighbour Field (NNF) algorithm. In 2013, Le Meur et al. [17] introduced the concept of hierarchical inpainting, in which the inpainting process is performed at multiple scales, beginning with a coarse scale and then refining the results on finer scales. At the coarse scale, the input image is reduced to a low-resolution image to reduce noise sensitivity and computational complexity, and a K-Nearest Neighbour (KNN) model is used to determine the priority selection of patches. Super-resolution is then used on the finer scale to increase the resolution of the image and recover high-frequency details using the information from the coarse scale as a basis, allowing the model to fill in the missing or damaged pixels at multiple resolution scales.

Bertalmio et al. [15] proposed a geometry-based method that uses Partial Differential Equations (PDEs) and diffusion. This model assumes that missing pixels can be predicted based on the values of neighbouring pixels and uses isophotes (lines of constant intensity) along PDEs to propagate local features from surrounding regions to damaged areas. Other researchers further developed the use of PDEs, including Chan et al. [18], who used total variation regularisation in combination with PDEs, Li et al. [19], who combined PDEs with smoothness constraints as regularisation techniques to improve the smoothness of the final result, and Telea [20], who proposed a Fast Marching Method (FFM) to simplify and speed up diffusion methods based on PDE.

Currently, traditional inpainting methods are still used due to their simplicity and low computational requirements, achieving good performance for small areas of missing or damaged pixels. However, when applied to larger areas and high-resolution or complex images, they often produce blurry or distorted results.

2.1.2 Deep learning methods

To address the limitations of traditional inpainting methods, artificial intelligence deep learning inpainting models were developed. The use of CNNs [21] trained with large image datasets was shown to have good results in learning how to fill in missing pixels in a more natural and realistic way. However, the most significant breakthrough in deep learning inpainting models came from the introduction of GANs in 2014 by Goodfellow et al. [22]. These networks consist of two parts: a generator and a discriminator. The generator's task is to create new data that are similar to the data it was trained on, while the discriminator's task is to determine whether the data it receives are real or generated by the generator. The two parts of the algorithm compete with each other, with the generator trying to produce data that are good enough to fool the discriminator, while the discriminator is trying to improve the identification of fake data. This competition drives the generator to improve, resulting in increasingly realistic outputs. Figure 2.1 illustrates the structure of the network.

In 2016, Pathak et al. [23] were the first to use GANs in an inpainting model, introducing the concept of Context Encoders (CEs). CEs are based on an Encoder-Decoder architecture (see Figure 2.2), with an encoder extracting feature representations from the damaged input image and the decoder processing the extracted features to restore the missing areas. The model employed two loss functions: a reconstruction loss to capture the overall semantic visual structures and the consistency of the repaired area with the surrounding visible area, and an adversarial loss to make the final result more realistic and sharper. The balance of these two losses is essential to generate good results.

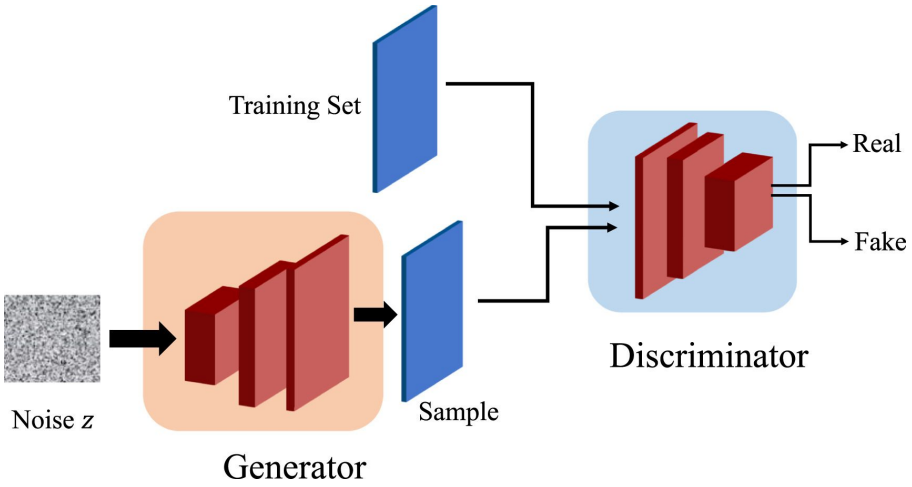

Figure 2.1: Explanation of a Generative Adversarial Network (GAN) [9]

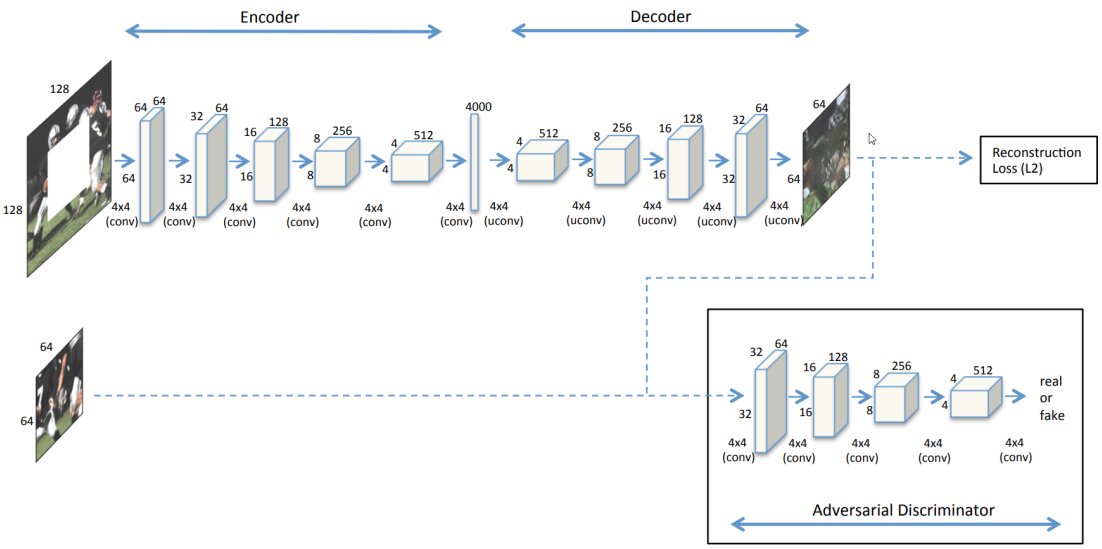

Figure 2.2: Example of a Context Encoder (CE) architecture [23]

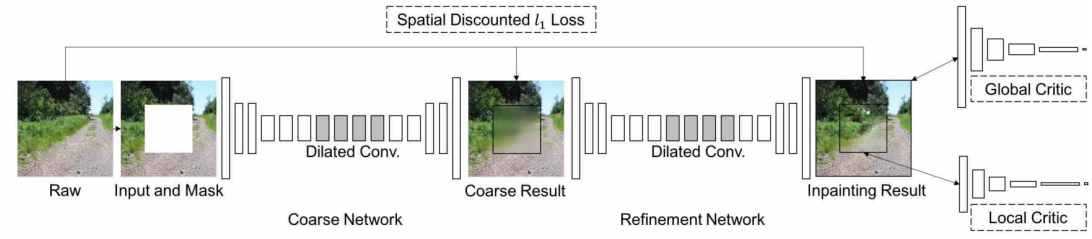

Figure 2.3: Example of a Course-to-Fine network [25]

This work served as the foundation for many more in the following years. In 2017, Iizuka et al. [24] improved the model by using two discriminators instead of one, a local and a global. The local discriminator concentrated on refining the area around the restored region, while the global discriminator refined the restored part with respect to the picture as a whole. A year later Yu et al. [25] proposed a new approach that improved the use of distant surrounding image features in the generation model, the course-to-fine network. The first stage generated a rough low-resolution result, and the second used a Contextual Attention Module (CAM) that employed the cosine similarity between the background and foreground feature patches to refine the result, as seen in Figure 2.3. CAM was designed to use a spatial discounted reconstruction loss to promote spatial coherency, combined with a local and global Wasserstein adversarial loss for more realistic results. The Wasserstein Generative Adversarial Network (WGAN) [26] is an improved version of the traditional GAN, which addresses the issue of potential instability and convergence during the training process by using a new loss function on the discriminator called the Wasserstein distance or Earth Mover's distance, as well as introducing the gradient penalty regularisation term. This model, known as DeepFillV1, produced better overall results with complex images, however, due to its two-stage process and complexity, it required a large amount of computational resources.

In 2018, Liu et al. [27] proposed the use of partial convolutional layers instead of fully connected convolutional layers to address the issue of flexibility in the size and shape of the input inpainting mask, which older models depended on regular square/rectangular masks. This was achieved using the U-Net architecture introduced by Ronneberger et al. [28]. This architecture consists of a contracting path that reduces the input image and an expansive path that reverts it to the original size. Skip connections are used to pass information from the contracting to the expansive path, allowing the network to use high-level and low-level features to make more accurate predictions with better performance. U-Net was initially developed for image segmentation tasks, but was also found to be suitable for inpainting tasks, as it allowed the model to handle masks of any size, shape, and location without compromising performance.

A year later, Nazeri et al. [29] proposed the EdgeConnect method to address the susceptibility of inpainting models to produce smooth and blurry results. This method focused on a two-stage process, edge detection and image completion, which mimics the way humans draw images, by first drawing lines and then colouring. The first stage involved an edge generator using an adversarial training approach to create edges that are difficult to distinguish from the ground truth edges. This generated a rough edge map of the missing region, which was then used as prior knowledge for the next stage to reconstruct the image based on the predicted edges. Figure 2.4 shows the architecture in more detail.

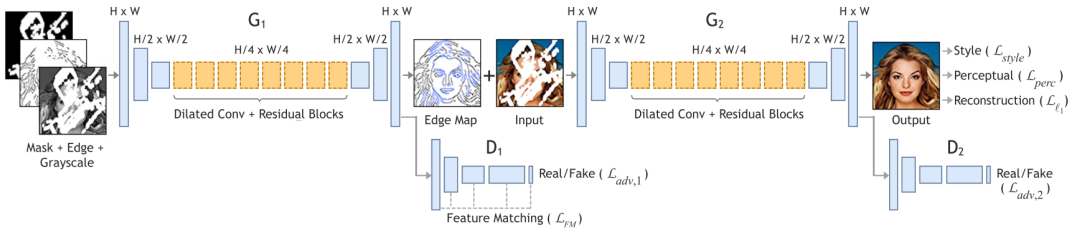

Figure 2.4: Architecture of the EdgeConnect model [29]

Taking into account the various advances in inpainting models, the team behind DeepFillV1 (Yu et al. [25]) decided to improve their model, taking cues from the work of Liu et al. [27] with partial convolutions and Nazeri et al. [29] with EdgeConnect, which resulted in DeepFillV2 [30]. The architecture used (Figure 2.5) was similar to the original, the main innovation being the use of gated convolutions, an improved method based on partial convolutions. The gated convolutions provide a learnable dynamic feature selection mechanism for each channel of each spatial location, allowing for an optional user-guided input to enhance the final result. Furthermore, the team employed Spectral Normalization (SN) [31] combined with PatchGAN [32], a GAN variant that applies the concept of traditional patch-based models and processes an image by dividing it into patches for the discriminator (SN-PatchGAN), which expedites model training.

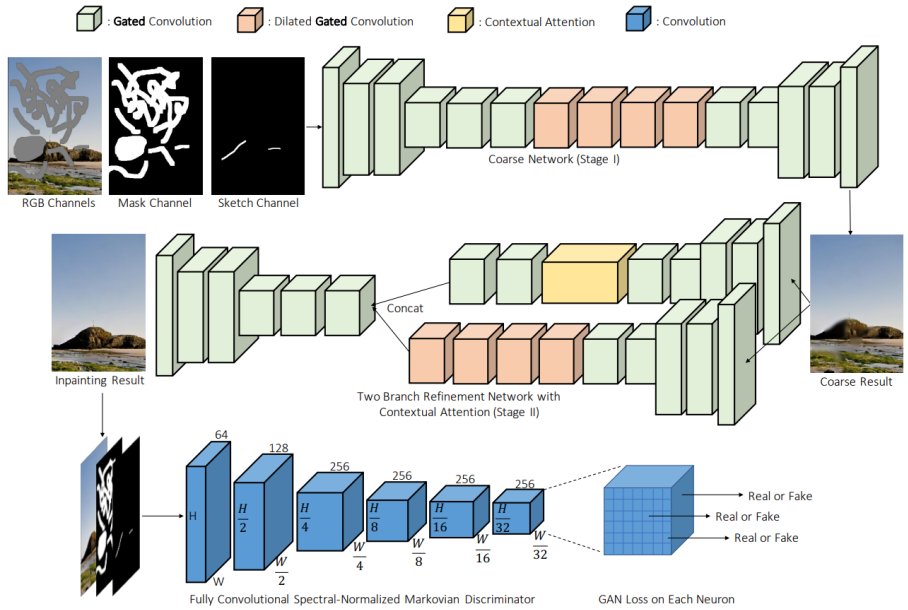

Figure 2.5: Overview of the network architecture for DeepFillV2 [30]

In addition to the model architecture, DeepFillV2 [30] also proposed an algorithm to automatically generate free-form masks to be used in the training process. This idea of incorporating irregular random masks had already been discussed in the work of Liu et al. [27], where they generated around 50,000 unique masks and during training randomly sampled one for each image, performing additional operations as well such as random dilation, rotation, and cropping. However, this method had several drawbacks: it was not efficient in storage and computation; the masks generated mostly did not reflect common use cases; and it was not

controllable or flexible. On the other hand, the new algorithm by Yu et al. [30] was incorporated into the training process, generating new masks on the fly without additional storage and with a small computational overhead. This was achieved by simulating the drawing of lines with rotation, similar to a brush or eraser movement, which also allowed for fine-tuning the shape and width of the strokes.

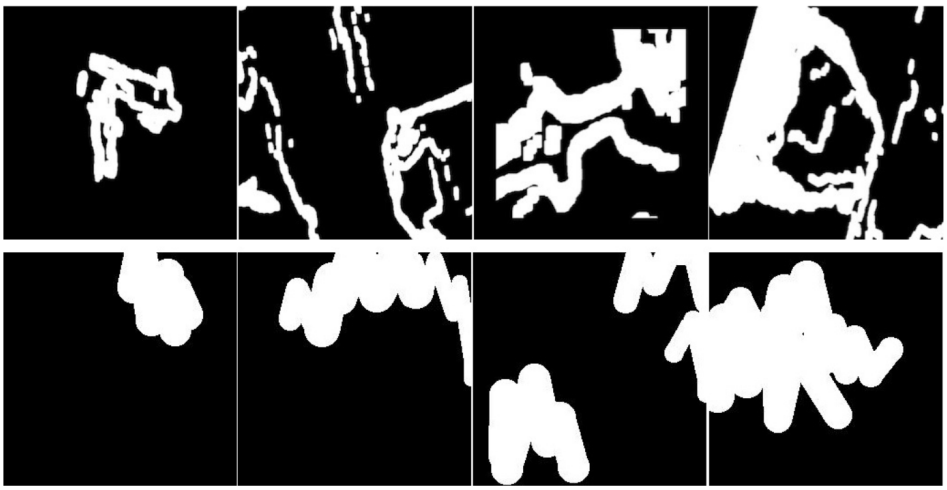

Figure 2.6: Comparison of training mask generation algorithms [30]. On the top are the results of Liu et al. [27] and on the bottom for Yu et al. [30].

This section provided a brief outlook on the evolution of inpainting models, from traditional to deep learning, and serves as a basis for the explanation of the current state-of-the-art models discussed in the following section.

2.2 Current deep learning inpainting models

The Computer Vision and Pattern Recognition (CVPR) is an annual event hosted by the Institute of Electrical and Electronics Engineers (IEEE) and the Computer Vision Foundation (CVF), renowned for being a premier conference in the field of computer vision and pattern recognition. It serves as a platform for researchers to present their work, discuss ideas, and stay informed of the most recent advances in the field. Additionally, New Trends in Image Restoration and Enhancement (NTIRE) workshops and challenges are held in conjunction with the conference to promote the development of new techniques and provide a unified platform to compare and assess various approaches to image restoration and enhancement.

In the 2022 edition, one of the challenges focused on image inpainting [33], with the aim of standardising a set of diverse and challenging image input masks and providing a benchmark consisting of different scene representations such as faces, objects, landscapes, and creative art. The challenge was divided into two tracks: the first was a fully unsupervised scenario, assessing the resulting image in comparison to the ground truth; and the second was a conditional scenario, taking into account an additional semantic input map and evaluating how the resulting image semantic labels compared to the input map. To address irregular mask shapes, seven of the most common types of masks were used in the training process. As depicted in Figure 2.7, from left to right these were: three types of strokes (thin, medium, and thick); image completion, ranging from 20% to 80%; uniform removal of every N pixel

lines; centre square image expansion; and nearest neighbour, removing N random neighbour pixels.

Figure 2.7: NTIRE 2022 image inpainting challenge masks [33]. The purple represents the masked region.

Taking into account that each solution needed to generate results from different types of scene representation, the following four datasets were chosen for the model training phase.

Flicker Faces High Quality (FFHQ) [34]: A public collection of 70,000 high-resolution images of human faces, collected and curated by researchers at Nvidia. It covers a wide range of ages, ethnicities, genders, and a variety of accessories, such as glasses and hats;

ImageNet [35]: One of the biggest public image datasets, started in 2009 and currently contains over 14 million annotated images of a variety of objects. It has been widely used over the years to train computer vision artificial intelligence models;

WikiArt [36]: A visual art encyclopaedia website. It features a large database of artworks and artist profiles, detailing accurate and reliable information about 250,000 artworks and 3,000 artists around the world. These artworks cover a wide range of media, including paintings, sculptures, drawings, prints, and photographs;

Places [37]: A dataset for scene recognition composed of more than 10 million annotated images covering more than 400 unique scenes. It was created to enable research on a wide range of computer vision tasks, including image classification, object detection, and scene recognition;

The performance of the inpainting models was evaluated by comparing the results with the ground truth using various objective and subjective metrics. The objective metrics included Peak Signal-to-Noise Ratio (PSNR) [38] and Structural Similarity Index Measure (SSIM) [39], which measure the similarity between two images taking into account objective quality measures such as the intensity and luminance of the pixels, with a higher score indicating better quality. The PSNR is determined by the ratio of the maximum signal power to the power of noise in the reconstructed image or signal. The SSIM is calculated by multiplying the difference of three terms: brightness or luminance; contrast; and structure.

Perceptual metrics are used to measure the similarity between two images from the point of view of a human observer, capturing the subjective perception of image quality. These metrics began to appear in the late 2010s and have become increasingly popular in recent years, allowing for a more accurate assessment of the performance of inpainting models. The two most commonly used are Fréchet Inception Distance (FID) [40] and Learned Perceptual Image Patch Similarity (LPIPS) [41]. The FID uses a CNN to extract features from both the

original and generated image, then calculates the Fréchet distance between the Gaussian distributions defined by the mean and covariance of the extracted features. The LPIPS calculates the average of similarity ratings between a set of patches from the original and generated image based on their feature representations from a trained CNN. In both metrics, a lower score is preferable.

According to the challenge report [33] more than 100 participants registered for this challenge, however, only the best performing teams were featured in the report. To establish the baseline metric values, the organisers trained two state-of-the-art models, CoModGAN [42] and LaMa [8], and used the evaluation metrics mentioned above to compare the model. The top three teams in the first track of the challenge all used a variation of either CoModGAN or LaMa, which further confirms the organisers' decision to use these two models as baselines.

The following sections of this chapter will provide more details on these two models, as well as Mask-Aware Transformer (MAT) [43], one of the finalists for the best paper at the 2022 CVPR conference, and Stable Diffusion [44], which gained popularity in 2023 for its innovative use of text prompts for image generation and editing.

2.2.1 CoModGAN

The GAN architecture initially proposed by Goodfellow [22] was a fully unsupervised model that could generate new data from random noise. However, this approach was complex to train and often produced samples without structure or coherency, which is not suitable for inpainting. To address this, Van et al. [45] proposed a variation of the GAN that used input data (e.g., image, class label, text) in addition to random noise to condition the generator. This allowed the results to be more consistent with the input data, but also increased the chance of collapse (the generator producing limited or repetitive output) and underperforming when the conditional information was limited. To overcome these issues, Zhao et al. [42] proposed a Modulated GAN, which used an unconditional architecture as the basis, but added a modulation layer to the generator. This layer was composed of a set of weights that could be adjusted based on the input data, providing some control over the results generated while still allowing the generator to maintain its capabilities.

In 2021, Zhao et al. proposed CoModGAN (Co-Modulated Generative Adversarial Network), a solution that combined the generative capability of unconditional modulated architectures with conditional input. This architecture was designed to generate diverse, yet consistent, images with regular and irregular inpainting masks, and to generalise to small-scale and large-scale inpainting with minimal conditional information. Additionally, two new metrics were proposed to evaluate the results of the model, Paired Inception Discriminative Score (P-IDS) and Unpaired Inception Discriminative Score (U-IDS). Similarly to FID, these metrics are calculated using a pre-trained CNN to extract features from the image, but are then fitted by a linear Support Vector Machine (SVM) to reflect the linear separability in the feature space. The P-IDS measures the probability that a generated image corresponds to its original image, while the U-IDS calculates the misclassification rate from a sample of real images to a sample of generated images, with both metrics evaluated as a higher being better. The authors of the article claim that these metrics have three main advantages over FID: better correlation with human preferences; better effectiveness in capturing subtle differences; and more robustness to the sample size.

Several experiments were conducted with the datasets FFHQ [34] and Places [37], using FID, U-IDS, and P-IDS as evaluation metrics for various input masking ratios. The average of five runs showed that CoModGAN outperformed the existing models at the time, DeepFillV2 [30] and PatchMatch [16], in all experiments.

2.2.2 LaMa

As described in section 2.1, DeepFillV2 [30] and EdgeConnect [29] have provided technical advances for deep learning inpainting models. However, these models often had a complex architecture with multi-step processes, while still having difficulty with large missing areas and high-resolution images. To address these issues, Suvorov et al. [8] proposed LaMa (an abbreviation for Large Mask) in 2021, a single-stage network inpainting model that can handle small and large missing areas, while supporting images with a resolution of around 2,000 pixels horizontally. To achieve these results, LaMa was developed with three novel concepts: a feed-forward network with Fast Fourier Convolutions (FFCs) [46]; a multi-component loss that combines adversarial loss and high receptive field perceptual loss [47]; and an aggressive training mask generation strategy.

The receptive field is the region of an input image that a neural network processes, which is determined by its central location and size. Conventional CNNs usually have a limited receptive field, meaning that the output is only affected by a localised region, and thus lacks spatial invariance. On the other hand, fast FFCs [46] have a receptive field that covers the entire image, allowing them to process both local and global contexts in parallel, making them more efficient in processing high-resolution images. FFCs replace the generator convolution layers when computing adversarial loss, and when combined with high receptive field perceptual loss [47], the focus is shifted to global consistency, reducing blurred results.

In addition, the LaMa architecture includes a new training mask generation strategy, which Suvorov et al. [8] suggest has a significant impact on the model's final performance. This approach involves generating masks composed of random samples from a combination of wide masks (polygonal chains dilated by random widths) and box masks (rectangles of various aspect ratios), up to a 50% ratio to the input image. This is a more aggressive approach than the one used by DeepFillV2 [30], with a focus on training with larger masks that generally improved performance for both narrow and wide masks. The full LaMa architecture is illustrated in Figure 2.8.

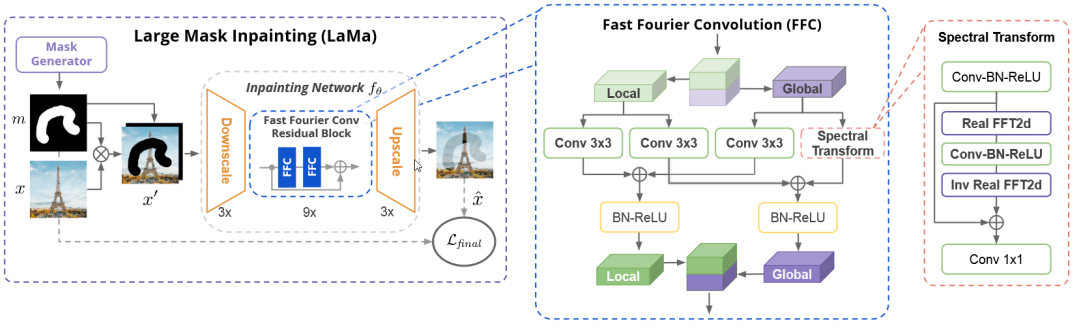

Figure 2.8: LaMa model architecture [8]

The model was tested with LPIPS [41] and FID [40] in two datasets, Places [37] and CelebA-HQ [48] (human face images similar to FFHQ [34]), narrow and wide masks. The results obtained were generally better than the baseline models (DeepFillV2 [30]; EdgeConnect [29]; etc.), but not as good as CoModGAN [42] and MADF [49] in narrow masks. However, LaMa achieved better results on average than these two models, while being more lightweight and efficient.

2.2.2.1 LaMa improvements

Following the development of LaMa [8], the same team proposed Big-LaMa. Contrary to the original model, which was based on efficiency and being lightweight, Big-LaMa has a higher number of FFC layers and was trained on a larger dataset. Despite its larger size, Big-LaMa still had fewer training parameters than CoModGAN [42] and MADF [49], and was able to surpass them in both narrow and wide masks.

Kulshreshtha et al. [50] proposed a technique called Multiscale Feature Refinement to improve the results of the LaMa model by reducing the blurriness of high-resolution images. This method follows a coarse-to-fine approach (see Figure 2.9) and iteratively adds more detail without the need to retrain the model. It replaces the image upscaler method, beginning with the training image resolution, which is assumed to be the more detailed version of the inpainting result. The technique then performs multiple feature refinement iterations, updating the model feature maps, and upscaling the image at each iteration. This method was tested in conjunction with Big-LaMa on 1,000-pixel-resolution images and was found to be more effective than the original model, although it took longer to infer. The LaMa team noticed the improvements made by this work and officially integrated it into their open-source repository.

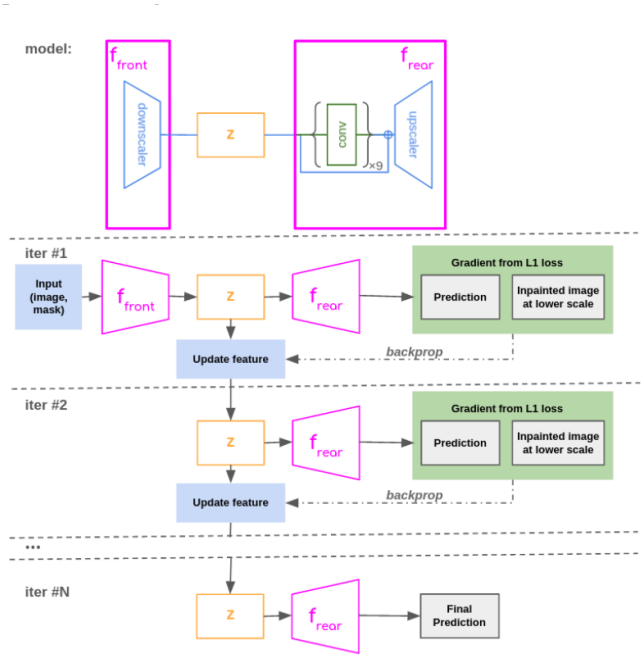

Figure 2.9: Multiscale Feature Refinement architecture [50]. The "front" and "rear" correspond to the encoder and decoder sections of the network.

The second best performing model on the unsupervised track of the NTIRE 2022 image inpainting challenge [33] was GLaMa, developed by Lu et al. [51] under the team name AIIA. This model improved on the original by introducing a mask generation strategy that supports seven different types, as shown in Figure 2.7, and a joint spatial and frequency loss, in addition to adversarial and high receptive field perceptual field loss, to regularise the optimisation process. The model was evaluated using the challenge specifications detailed at the beginning of this section (section 2.2), and achieved more accurate results in all metrics compared to the base LaMa, without changing the core architecture.

2.2.3 MAT

Li et al. [43] proposed MAT, a large mask inpainting model based on Transformers. This type of neural network was first proposed by Vaswani et al. [52] in 2017 and is specialised in the processing of sequential data. It is characterised by its self-attention mechanism, which allows the model to dynamically weigh the input data and focus on the most relevant parts of the sequence for a given task. This is achieved by using multiple attention heads that are trained to attend to different parts of the input data. Transformers have become a popular choice for natural language processing tasks, but they have also been increasingly used for computer vision tasks. MAT was proposed as an alternative to existing state-of-the-art models such as LaMa [8] and CoModGAN [42], which were based on GANs.

The MAT architecture, depicted in Figure 2.10, consists of a convolutional head, a transformer body with blocks of varying resolutions, and a convolutional tail with a high-frequency detail refinement network. The convolutional head processes the input image and mask, which extracts feature tokens, and downscales the data to reduce computational complexity. The transformer body then uses multi-head self-attention to learn contextual information from the tokens and reconstruct the missing areas. Finally, the convolutional tail refines the details of the reconstructed image and upscales it to the original resolution.

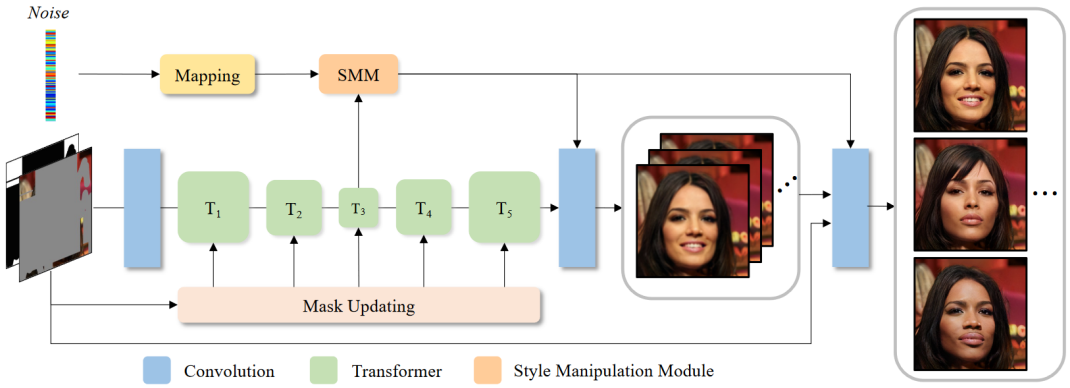

Figure 2.10: Mask-Aware Transformer (MAT) architecture [43]

The model was tested using Places [37] and CelebA-HQ [48] datasets and the FID, U-IDS, and P-IDS metrics. It achieved better results than other state-of-the-art models in the CelebA-HQ dataset. However, compared to CoModGAN [42] and LaMa [8] in the Places dataset, it performed better on some metrics and worse on others.

2.2.4 Stable Diffusion

In 2022, Rombach et al. proposed Stable Diffusion, a text-to-image deep learning model. This model is based on diffusion techniques [53] and is used mainly to create detailed images from text descriptions, but it can also be adapted to other tasks, such as inpainting. Diffusion models are generative models that work by adding Gaussian noise to the data and then learning to reverse the noising process to generate new samples, as shown in Figure 2.11. To generate a specific result, this process can be guided using conditional information such as text embeddings at each step.

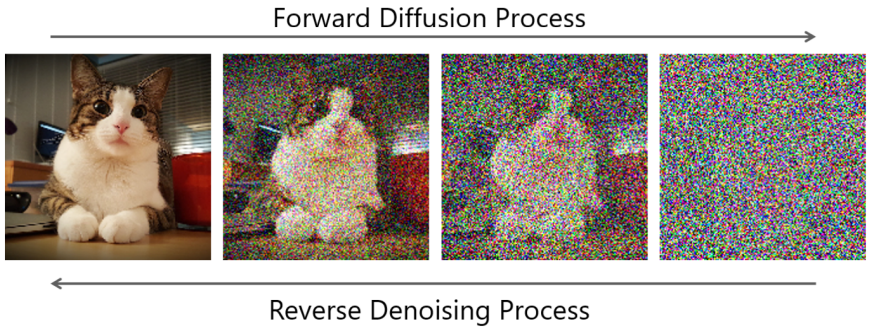

Figure 2.11: Diffusion model image generation process [54]

Stable Diffusion uses a Latent Diffusion Model (LDM), a variation of the base diffusion model that uses a latent space instead of a pixel space, which contains only the most essential information from the original data. The use of a more compressed space allows the models to be more efficient in the training process and to be able to accept both images and text as conditional input. The LDM is based on an Encoder-Decoder architecture and begins the generation process by encoding both the input image and the conditioning inputs into the latent space. These two are then combined using an attention mechanism that learns the best way to combine the two in latent space. The data is then decoded using a U-Net [28] network to generate the final result. This architecture is illustrated in Figure 2.12.

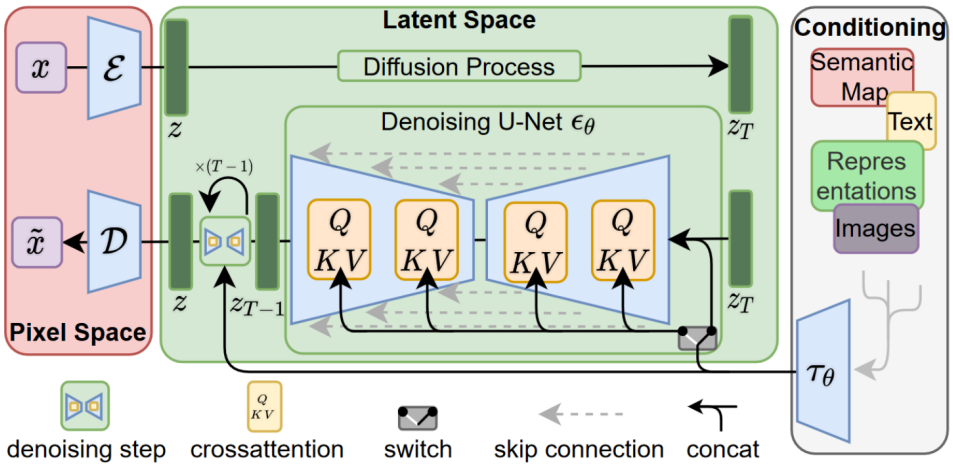

Figure 2.12: Stable Diffusion model architecture [44]

The authors tested the model as an inpainting model using the same methodology as LaMa [8] and compared it to the base LaMa model. The results revealed that Stable Diffusion had a better FID [40] but a worse LPIPS [41] than LaMa.

The popularity of text-guided image inpainting increased in 2023 after the introduction of Stable Diffusion as an open source solution, with new solutions created, such as Smart Brush by Xie et al. [55] and Imagen Editor by Wang et al. [56]. By providing a text prompt, users can not only remove objects from images but also replace them with something else. However, when it comes to pure image object removal, the use of a text prompt is a disadvantage, since the model relies on a conditional text description and produces random results without it.

2.3 Current application fields of inpainting

Inpainting techniques can be used for a range of image manipulation tasks. According to the research of several papers [9–12], the most common applications are as follows.

- Restoration of old photos by filling in missing or damaged parts of the image and repairing scratches;
- Removal of unwanted objects like trees, cars, people, and animals;
- Improvement of image quality and upscaling;
- Text removals like subtitles and product labels;
- Creation of digital art by removing regions and filling them with unique elements;

Looking at real-world applications, some examples of works that are used inpainting models include the work of Wan et al. [4, 5] on the restoration of old photographs, Bartl et al. [7] on self-checkout systems, Pirnay and Chai [57] on image anomaly detection, and Xie et al. [6] on Japanese comic language localisation.

2.3.1 Restoration of old photographs

In contrast to modern pictures, the restoration of old photographs is more difficult due to the disparity between synthetic and genuine photos, the various kinds of flaws present (e.g. film grain, sepia effect, colour fading, scratches), and the emphasis on preserving facial intrinsic features since a large portion of old photos are portraits.

Wan et al. [4, 5] proposed a solution that used two Variational Autoencoders (VAEs) [58] and leveraged data from a combination of old real photos, synthetic data degraded from modern clean images, and the corresponding ground-truth modern images. VAEs are designed to learn a continuous latent space, which is a low-dimensional representation of the input data. This enabled training one VAE on old degraded images and the other on clean modern images. Using these two models, the solution can learn the mapping between the two latent spaces. Figure 2.13 shows some restoration examples of their solution.

Figure 2.13: Old photos restoration examples by Wang et al. [4, 5]

2.3.2 Self-checkout systems

The use of self-checkout systems has become increasingly popular in recent times, providing customers and retailers with a number of advantages. These systems enable customers to scan and pay for their items without the need for store personnel. However, there is still a requirement that staff are present to assist when the process fails.

To further improve these systems, Bartl et al. [7] developed PersonGONE, a solution that uses a camera to detect and track the products being scanned. This solution provides assistance during the checkout process by verifying the products being scanned and providing an alternative method of identification in the event that a barcode is not read correctly. When developing their solution, the authors encountered a major issue that human hands were constantly present in conjunction with the products, which affected the performance of their object detection model. To address this, they implemented an auxiliary inpainting model to remove the customers' hands and focus solely on the products. LaMa [8] was used as the inpainting model, which processed the frame-by-frame video feed using a mask generated by the detection of human hands by the object detection model. According to the authors, the results obtained were satisfactory for a proof of concept, but the inpainting model processing each frame increased prediction delays and concluded that a more video-focused inpainting solution was needed to resolve this issue.

2.3.3 Image anomaly detection

The task of image anomaly detection involves recognising unusual patterns or deviations from the norm in images. It is commonly used to detect flaws in manufacturing processes or abnormalities in medical images. Deep learning inpainting has been used to detect anomalous patterns in images by attempting to fill in the anomalous parts of the image using a trained neural network. If the neural network is unable to fill in the anomalous parts of the image in a way that is consistent with the rest of the image, it can be an indication that the image contains an anomaly that needs to be examined further.

Pirnay and Chai [57] proposed InTra, a transformer patch-based inpainting model, as an example of an anomaly detection solution. This process begins by dividing the image into several square patches and adding positional embeddings to incorporate spatial context. The model then reconstructs each patch based on the information from neighbouring patches and calculates a pixel-wise anomaly score by comparing the result with the original image. Figure 2.14 shows some results from their solution.

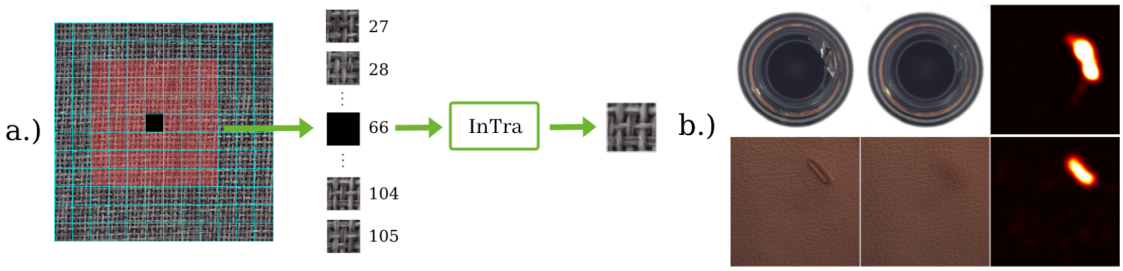

Figure 2.14: Examples from the InTra anomaly detection model [57]. On the left (a) is an example of how a black patch is generated by its neighbours, and on the right (b) is a comparison between the original and the generated image.

2.3.4 Japanese comic localisation

Manga is a style of Japanese comic book and graphic novel that encompasses a broad range of genres, such as action, adventure, etc., for readers of all ages. Manga publishers are an important component of the Japanese publishing industry, with millions of manga titles sold each year. In recent years, manga has become increasingly popular outside of Japan, leading to the emergence of localisation publishers that translate the content into other languages. However, due to the way manga is drawn, it often requires manual erasing and redrawing of elements such as speech bubbles and "sound effects" text.

Xie et al. [6] proposed a manga inpainting solution to make localisation easier. Their solution is composed of a two-step process with two inpainting models and is capable of seamlessly removing parts from manga pages. The first step involves decomposing the input image into a representation of the structural lines and the components of the screentone. This information is then fed into a VAE-based inpainting model to predict the semantic elements in the missing regions. The resulting semantic maps are then used as a correlation guide in the second step, with another model filling the missing regions by borrowing features from the known surrounding regions. Figure 2.15 shows an example of the results obtained by their solution.

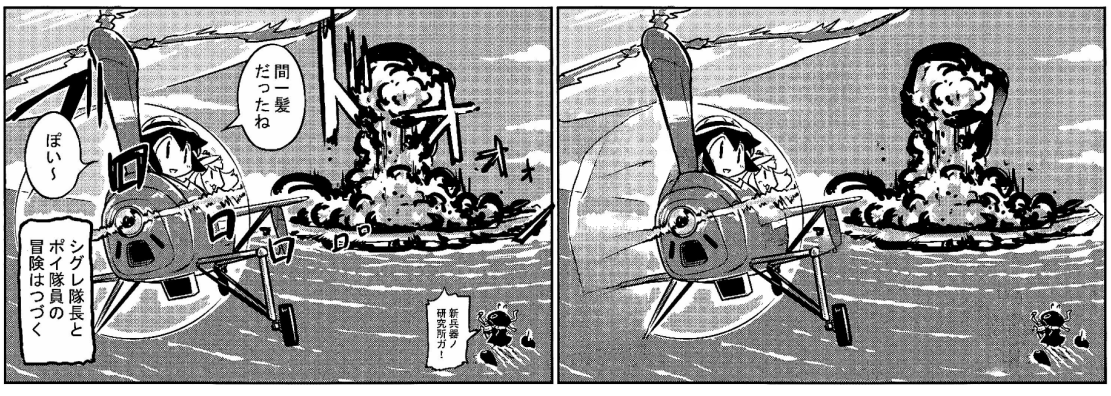

Figure 2.15: Manga inpainting examples by Xie et al. [6]. On the left is the original comic and on the right is the inpainting result without speech bubbles.

2.4 Existing image object removal solutions

In this section, eight image inpainting object removal solutions are discussed [2, 3, 59–64]. Table 2.1 presents a breakdown of the solutions, four of them tailored to the removal of objects from real estate images and the other four for general images.

Table 2.1: Analysis of image inpainting object removal solutions

Solution	Pricing	Purpose	Type
Styldod [59]	Paid	Real Estate	Service
Plan It All [3]	Paid	Real Estate	Service
Phixer [2]	Free Trial / Paid	Real Estate	Service
Revivoto [60]	Paid	Real Estate	Service
Cleanup Pictures [61]	Free [1] / Paid	General	Tool
Magic Eraser [62]	Free [1] / Paid	General	Tool
Adobe Photoshop [63]	Free Trial / Paid	General	Tool
Inpaint [64]	Free [1] / Paid	General	Tool

[1] Limited to low-resolution images

The eight solutions analysed can be divided into two categories: service-based and tool-based. Service-based solutions such as Styldod [59] and Phixer [2] provide real estate object removal as a service, where users upload images they want to edit and then mark the areas to be removed with an optional note description for each image. These edits are then ordered, with the results being delivered at a later time, depending on the chosen time frame, which can range from 12 to 48 hours. The pricing for this type of solution is usually based on the number of images being edited and the speed at which the user wants the results, although some solutions also offer monthly flat rates. In addition to object removal, these solutions also offer other common services for real estate images, such as day-to-dusk, 3D rendering, 360º virtual tours, and other image-editing enhancements. Since these types of solution require a large amount of time to process and may require additional user input, it is likely that the object removal process is either fully or partially manual.

Tool-based solutions such as Inpaint and Magic Eraser offer users full control over the object removal process. Artificial intelligence inpainting models are used to remove objects from the selected regions in real time. These solutions are usually free to use, although there are limits on the number and resolution of images, with a one-time or monthly payment needed to get rid of these restrictions. The main features of these solutions can be summarised in the following list.

- Image upload and download, supporting common image formats and several resolutions;
- Brush-based mask region selection, with options to change brush size and shape;
- History of each change, with the option to undo and redo;
- Comparison between the current result and the original image;
- Common image editing tools, like rotate and crop;
- Object removal of the selected region, either triggered by a button or immediately after selection;

2.5 Final discussion

In the beginning of this chapter, four questions were defined to provide a better understanding of past and current inpainting technologies, and each question was answered in the sections of this chapter.

The first section (section 2.1) outlined the evolution of inpainting models, which began with Efros and Leung [14] in 1999 and Bertalmio et al. [15] in 2000 and their traditional patch-based and diffusion-based approaches. Several improvements to their original models were made in the following years, but the most significant leap occurred with the development of deep learning models and the introduction of GANs by Goodfellow et al. [22] in 2014. From this point on, several deep learning inpainting models were created using GANs, such as CE by Pathak et al. [23] in 2016, DeepFill by Yu et al. [25] in 2018, and EdgeConnect by Nazeri et al. [29] in 2019.

The current state-of-the-art deep learning inpainting models were discussed in section 2.2. CoModGAN [42] combines the benefits of unconditional modulated generative architectures with conditional input, while LaMa [8] uses a single-stage network with FFCs and an aggressive training mask generation algorithm. These models have been widely used and have achieved the highest scores in the NTIRE 2022 image inpainting challenge [33]. MAT [43], one of the best paper finalists of the CVPR 2022 conference, is a newer model that uses transformer blocks and a multi-head self-attention layer to learn contextual information instead of GANs. Stable Diffusion [53] is a deep learning diffusion model and uses a LDM architecture to generate images from text descriptions. It is capable of not only removing objects from images, but also replacing them with something else.

In section 2.3, the application fields most commonly used for inpainting models are detailed, such as image restoration, image object removal, and text removal. Additionally, examples of past works that used inpainting models in their solution were detailed, such as the work of Wan et al. [4, 5] on the restoration of old photographs, Bartl et al. [7] on the removal of human hands for self-checkout systems, Pirnay and Chai [57] on image anomaly detection, and Xie et al. [6] on the removal of Japanese comic regions for language localisation.

The previous section section 2.4) discussed eight existing real estate object removal solutions available in the market. These solutions can be divided into service-based and tool-based. Service-based solutions such as Revivoto [60] and Plan It All [3] provide real estate object removal as a service. The user is required to pay for the removal of certain areas in the image, with the results delivered at a later time. Tool-based solutions such as Magic Eraser [62] and Cleanup Pictures [61] allow the user to remove areas of an image in real time,

by using deep learning inpainting models. This type of solution is more in line with the objectives of the project.

In the next chapter, the process and development of the proposed solution are detailed.

Chapter 3

Solution development

This chapter outlines the development of the proposed solution. It is divided into the analysis of the requirements, architecture design and deployment, and the creation of the solution's components. Additionally, it covers the social aspects of the solution, such as data protection and ethical considerations.

3.1 Requirements analysis

By examining the objectives of the proposed solution (outlined in section 1.3) and the features of existing solutions (discussed in section 2.4), a list of functional and nonfunctional requirements was created.

- The user should be able to upload an image to process, supporting common image formats and several resolutions;
- The user should be able to see the image;
- The user should be able to "draw" regions in an image to indicate the inpainting regions. This is done using a circular brush-based tool and the user should be able to distinguish between the selected regions and the image;
- The user should be able to change the size of the brush tool;
- The user should be able to process the inpainting of the image with the click of a button, which will send the information to the inpainting model;
- The user should be able to download the resulting image;
- The user should be able to redo and undo the results. This means that the solution keeps an image history of every change;
- The user should be able to compare the resulting image with the original image side by side;
- The solution needs to integrate with the Maxwork portal;
  - The user should be able to select an image from one of its listings to remove objects;
  - The user should be able to save the resulting image replacing the original;
- The solution should be responsive to different resolutions;
- The solution should support multiple languages;

Other requirements were also taken into account, however, due to the limited time available, they were deemed to be of lesser importance and not essential.

- The user should be able to clear the selected region;
- The user should be able to change the colour and shape of the brush tool;
- The user should be able to perform basic image editing functions, such as rotate, crop, and zoom;
- The solution should be compatible with mobile devices;

3.2 Architecture and deployment

The proposed solution includes a web application and an Application Programming Interface (API) to access the inpainting model. This web application, which is accessible through the Maxwork portal, allows users to interact with the solution and remove objects from images. The API provides an endpoint for using the model to predict the resulting image. To ensure scalability and consistency, a Docker container image of API was created.

The two components are hosted on Microsoft Azure cloud services, using the Web App and Container Instance services. The decision to use Azure was based on personal preference, though other services such as Amazon Web Services (AWS) are also viable. Figure 3.1 illustrates the architecture in a high-level overview.

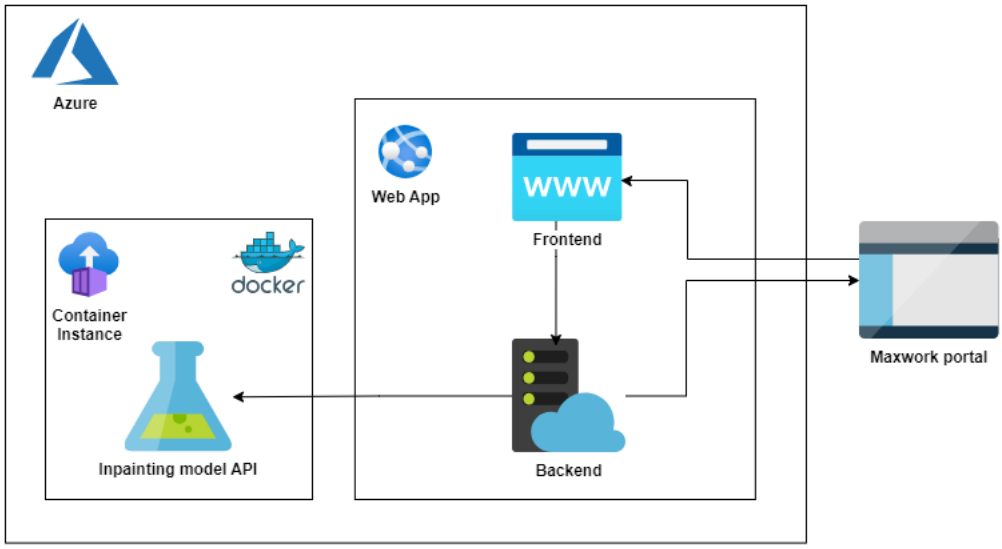

Figure 3.1: Proposed solution architecture

3.3 Inpainting model API

An API was developed to use the deep learning inpainting model, using FastAPI [65], a Python [66] web framework to build APIs.

The API has a single endpoint (Listing 3.1) that requires an authentication key and takes two images as input. The first image is the one to remove objects and the second is the selection mask. To prevent resource exhaustion or denial of service attacks, the size of both images is checked against a maximum size and then resized to a more manageable size for the model. The maximum size, as well as other constant values, can be configured through command-line arguments and environment variables in the docker image.

After loading both images, the inpainting model is loaded, and the result is predicted using the images as input. The model is loaded using TorchScript [67], a component of the PyTorch [68] machine learning framework that enables the serialisation and optimisation of PyTorch models for improved performance and deployment.

```python
@app.post("/inpaint")
async def inpaint(image_input: UploadFile, mask_input: UploadFile,
    api_key: APIKey):
    try:
        config = load_config()

        # load image inputs
        image = load_image(image_input)
        if validate_image_size(image, config.file_max_size):
            raise HTTPException(HTTP_400_BAD_REQUEST, "Input image size
    exceeds limit")
        image = resize_image(image, config.image_resize_max)

        mask = load_image(mask_input, gray=True)
        if validate_image_size(mask, config.file_max_size):
            raise HTTPException(HTTP_400_BAD_REQUEST, "Mask image size
    exceeds limit")
        mask = resize_image(mask, config.image_resize_max)

        # use inpainting model
        model = load_model(config)
        image_result = model.predict(image, mask)

        return Response(content=image_result, media_type=config.
    mime_type)
    except Exception as e:
        return HTTPException(HTTP_403_FORBIDDEN, "Internal Error")
```

Listing 3.1: Inpainting model API predict endpoint

3.3.1 Model selection

To select the model to use in the API, an experiment was carried out to validate the performance of three state-of-the-art models present in section 2.2. A sample of around 2000 images from the Places [37] dataset was used, which included a mixture of indoor and outdoor images to reflect the use case of real estate images. Medium-sized masks were

generated using the LaMa [8] mask generation algorithm and the results were compared using four evaluation metrics.

Table 3.1: Comparison between state-of-the-art inpainting models

Model	LPIPS ↓	FID ↓	U-IDS ↑ (%)	P-IDS ↑ (%)
CoModGAN [42]	—	—	—	—
LaMa [8]	**0.0836**	**7.2353**	**11.98**	**20.24**
MAT [43]	0.0948	7.9927	10.48	19.19

Unfortunately, despite multiple attempts to use the CoModGAN model to make predictions, it was unsuccessful, and as such only the results from LaMa and MAT are displayed in Table 3.1. However, taking into account the results obtained in the NTIRE 2022 image inpainting challenge [33] for CoModGAN and LaMa, CoModGan presented slightly better results in the Places dataset, but with a minor difference that could be attributed to the margin of error.

The results of the experiment showed that LaMa performed slightly better than MAT in all metrics. These results, and the fact that the NTIRE 2022 inpainting challenge report [33] indicated that FFCs networks have advantages in dealing with repeated patterns or textures, which are common in property images, led to the decision to use LaMa as the inpainting model.

3.3.2 Average response time

The prediction endpoint of the inpainting model API is highly dependent on the average response time, which can significantly affect the user experience and the overall performance of the system. To evaluate the average response time, 8 sets of 10 square images of varying sizes were used to call the prediction endpoint, half using Central Processing Unit (CPU) and the other half using Graphics Processing Unit (GPU). The hardware used for this test was a Ryzen 7 3700X CPU and a GeForce RTX 3070 GPU.

Table 3.2: Average response time of the inpainting model API in seconds

Device	256x256 px	512x512 px	1024x1024 px	2048x2048 px
CPU	0.31517 s	1.03427 s	4.87384 s	30.42631 s
GPU	0.06065 s	0.07539 s	0.28957 s	1.14361 s

Examining the data in Table 3.2, it is evident that GPU is much faster than CPU, with a performance several times faster. When comparing image sizes, the average response time increases exponentially, with the best user experience achieved with images up to 1024 pixels.

3.4 Web application

Since one of the objectives of the project (section 1.3) was to integrate with the Maxwork portal, the web application was developed using a template provided by the company that already included the authentication between the two applications. This template has a React library [69] for the front end and a .NET platform [70] for the back end. The structure of the web application is depicted in Figure 3.2, which follows a variation of the clean architecture [71].

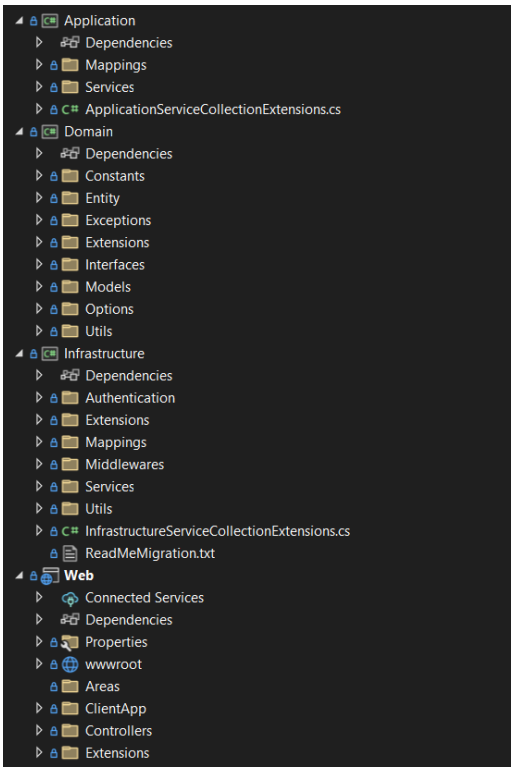

Figure 3.2: Web application structure

Following the requirements specified in section 3.1, the we application was first developed for public use for testing purposes and then modified to integrate with the Maxwork portal. In the public-access version, the application contains a simple landing page that redirects to a page where the user can upload an image by selection or drag and drop, as depicted in Figure 3.3.

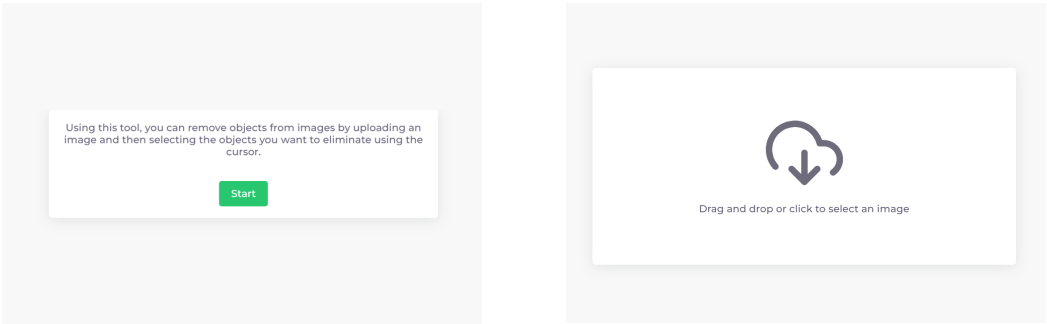

Figure 3.3: Web application landing and upload page

After selecting the image, the user can use the inpainting editor to remove objects from the image by selecting an area with the cursor and pressing the removal button. The application communicates with the inpainting model API by sending the image and the selection mask, then replaces the image in the editor with the result received. This editor, shown in Figure 3.4, also contains several features to enhance the user experience, as specified in the requirements of section 3.1. These features, from left to right, are the following:

- Change the selection brush colour;
- Change the selection brush size;
- Help modal with information about the application and its features;
- Reset the zoom level;
- Clear the current selection area;
- Undo the most recent selection or object removal;
- Redo the previous selection or object removal;
- Compare the original image with the most recent object removal result;
- Download the most recent object removal result;
- Remove objects from the image using the selected area;
- Save the most recent object removal result to the Maxwork portal. This option is only available when using the application from the Maxwork portal.

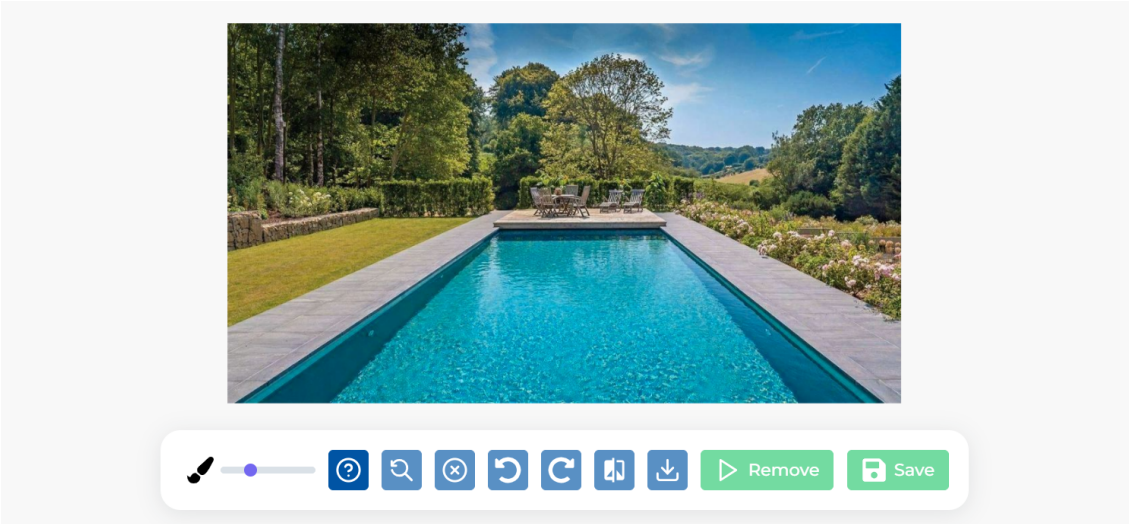

Figure 3.4: Web application inpainting editor page

3.4.1 Maxwork portal integration

The next step of development was to modify the application to be compatible with the Maxwork portal, allowing real estate agents to use the application in their current workflows. Figure 3.5 displays the page on the Maxwork portal where real estate agents can manage the images of a property they are listing. An additional choice was added to this page that leads the user to a new page that displays the inpainting editor as seen in Figure 3.4 with the

chosen image. To keep the user within the Maxwork portal, the object removal application is embedded into the portal through an inline frame (iframe), instead of being directed to an external link. When the user is satisfied with the result, it can click the save button to save the inpainting result to the Maxwork portal, replacing the original image, and redirecting the user back to the property image editing page.

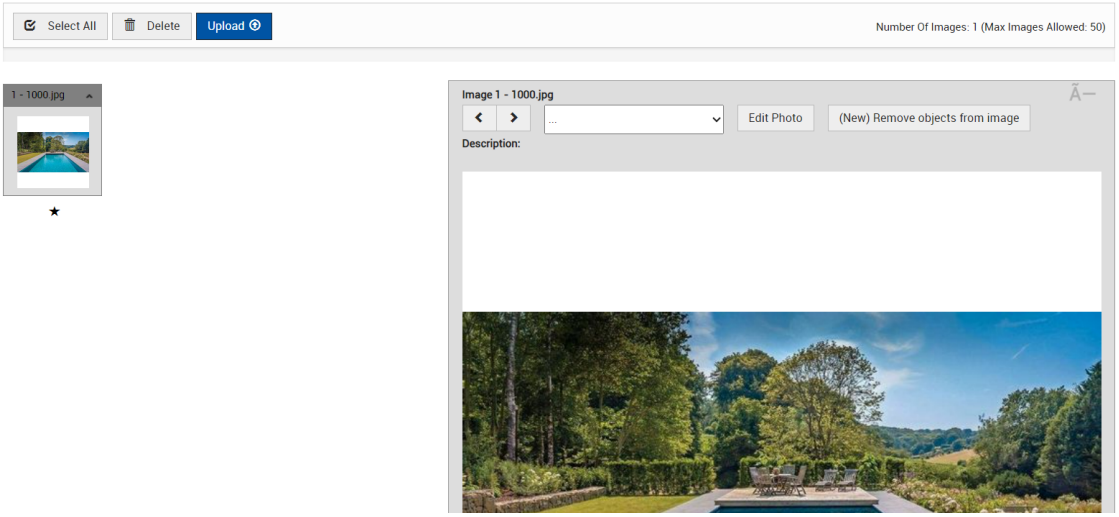

Figure 3.5: Maxwork portal property image editing page

3.4.2 User feedback

A feedback form (illustrated in Figure 3.6) was created to subjectively evaluate the application, based on anonymous user feedback. This form is optional and rates the application on a scale of 1 to 5 in terms of how easy it is to use and how satisfied users are with the results of the model. A public version of the application was made available to a small group of people and received a total of 16 reviews. The results were very positive, with an average rating of 4.31 for ease of use and 3.90 for satisfaction with the model result.

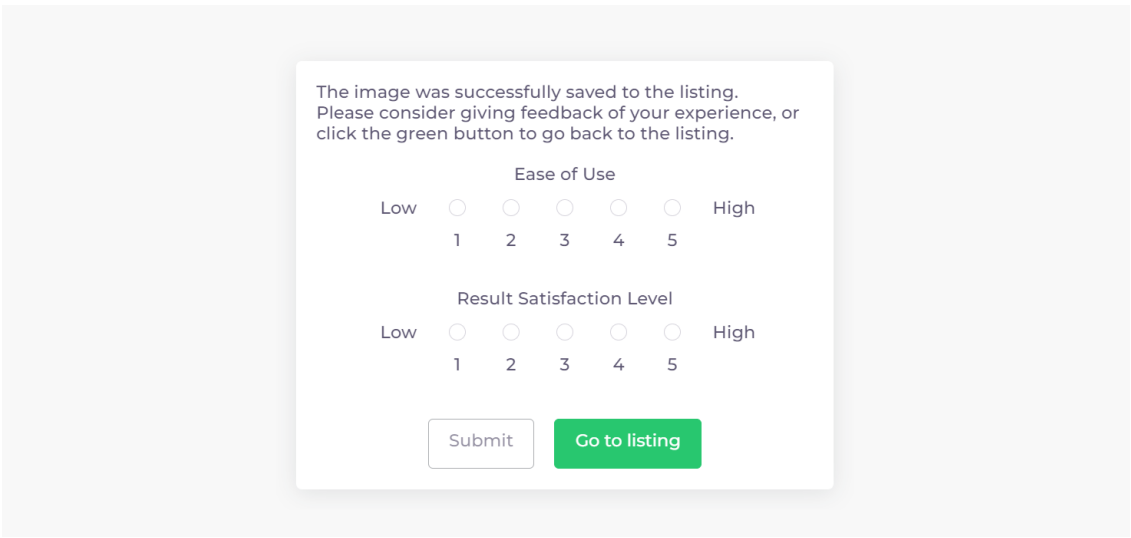

Figure 3.6: Web application feedback page

3.5 Social aspects

This section describes the social aspects of the proposed solution, namely, data protection, security, and ethical implications.

3.5.1 Data protection and security

The proposed solution is designed to manage both indoor and outdoor images of properties, which in some cases may contain personal data. To ensure compliance with the European General Data Protection Regulation (GDPR), the existing regulations of the RE/MAX Portugal Maxwork portal will be applied, since the proposed solution will be integrated with it. This includes, but is not limited to, obtaining consent from individuals before processing their data, implementing appropriate technical and organisational measures to protect the data, and providing individuals with access and the ability to delete their data upon request. Access to these data is secured by user credentials and is only available to the user and RE/MAX representatives, with public access if the user consents.

3.5.2 Ethical and moral implications

The use of image editing tools by real estate agents has complex moral and ethical implications that can be seen from different angles. Some people contend that editing images to present a more attractive or desirable view of a property is a form of deception and can be considered morally wrong. This is especially true if the changes are significant enough to alter the overall look of the property or if they conceal flaws or issues that a potential buyer should be aware of without disclosing them.

Others argue that editing images is an essential part of marketing and promoting a property. Real estate agents and photographers often use editing techniques to improve lighting, eliminate distractions, and emphasise the best features of a property. They argue that these edits are not meant to deceive but rather to present the property in the most favourable light. These changes are mostly cosmetic and do not affect the main characteristics of the house.

From a legal point of view, most countries have a law that requires real estate agents to be honest and transparent when advertising a property and not to make any false or deceptive statements that have an effect on a permanent item, fixture, or issue that cannot be easily fixed. Consequently, it would be unlawful to alter pictures in a way that conceals significant information or misrepresents the property's characteristics such as size, features, fixtures, and location.

In conclusion, when it comes to real estate images, it is essential that agents take responsibility for ensuring that the images accurately reflect the property. Any edits made should not be intended to conceal defects or issues, and any major changes should be disclosed to potential buyers. Editing real estate images can be a beneficial tool for marketing and promoting a property, but it is important to make sure that the edits are not deceptive or misleading.

Chapter 4

Experimentation and results

The LaMa pre-trained model used in the proposed solution was trained with images from the Places [37] dataset. This model is suitable for real estate image object removal, as the dataset contains images from both indoor and outdoor settings, but it also contains a large number of images that are not applicable to real estate. This chapter describes an experiment conducted to train an inpainting model using a dataset composed of real estate images previously used on sold listing, and how it performed compared to pre-trained LaMa model. The experiment was carried out using open source code available on the LaMa GitHub repository [72].

4.1 Dataset

The RE/MAX Portugal real estate company supplied a collection of images of listings that had been sold. This dataset was made up of approximately 100,000 high-quality images of the interior and exterior of the properties. Figure 4.1 shows a sample of the dataset.

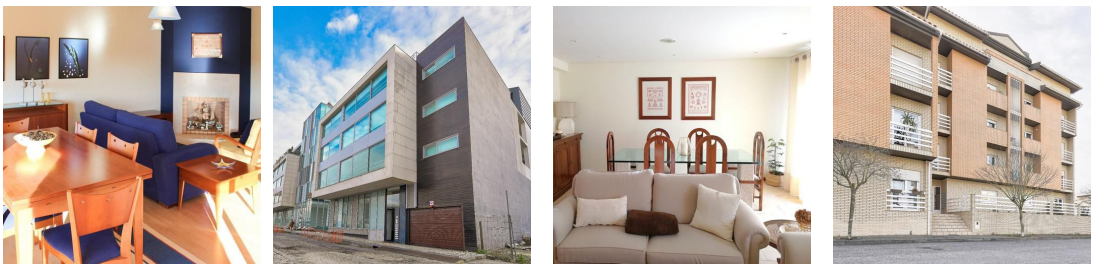

Figure 4.1: RE/MAX Portugal real estate image dataset sample

The dataset was prepared for the model by resizing the images and randomly cutting them into a 1024 by 1024 pixel square shape. An holdout technique was then used to split it into a 70% training, 15% validation, and 15% test subset, allowing the model to be tested on data that had not been seen before. The last step was to create random masking images for the validation and test subset to guarantee the consistency of the evaluation results between models. The LaMa mask generation algorithm detailed in subsection 2.2.2 was used to generate masks of up to a 50% ration of the input image. The training subset did not need this step since a new masking image is generated at each step of the training process.

4.2 Model training

Two approaches were considered for training the new model: transfer learning and training from scratch. Transfer learning involves using a pre-trained model as the starting point and reusing the learnt features from the pre-trained model. This was the initial option considered as it allowed the LaMa model to be adapted to real estate images. Despite numerous attempts, it was not possible to use the weights of the pre-trained model, so the decision was made to train the model from the beginning.

The architecture of the model was set up with the same specifications as the pre-trained model, consisting of 18 FFC residual blocks and an Adam optimiser with a fixed learning rate of 0.001 for the generator and 0.0001 for the discriminator. The model was trained with a batch size of 2 for 20 epochs, each with 20,000 steps. This batch size was reduced from the original 120 due to hardware restrictions, as only a single GeForce RTX 3070 GPU was used. The complete configuration is shown in Listing A.1 in Appendix A, and a more detailed explanation of the LaMa architecture is provided in subsection 2.2.2 and in the original LaMa paper [8].

At the end of each epoch, a metric was used to assess the model using the validation dataset to identify the best epoch of the training process. This metric was a combination of the L1 and L2 distances at the pixel level, along with the LPIPS and FID metric. Figure 4.2 shows the validation metric by step graph plot.

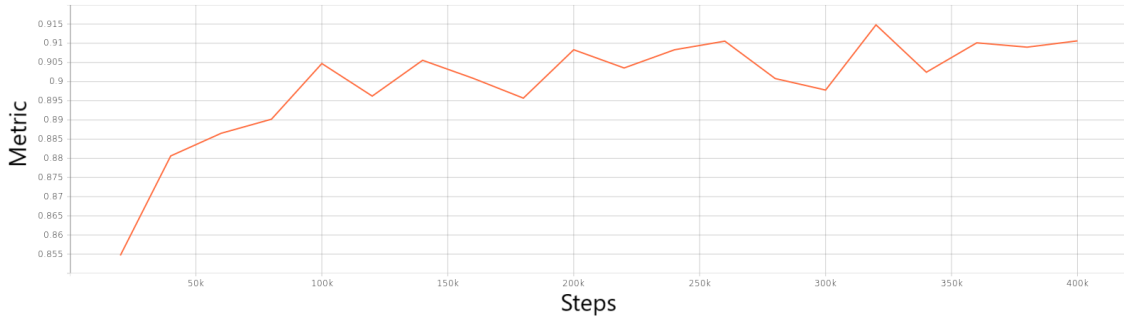

Figure 4.2: Trained model validation metric by step plot

Examining the graph, it reveals a regular pattern of rises and dips every two epochs. This is likely due to the batch size used, and a larger size could result in a smoother plot line. The highest performing model was at step 320,000, which is the 16th epoch, and performance steadily declined after this point.

4.3 Model evaluation

To compare the performance of the new trained model with the pre-trained LaMa model, both were assessed using the test subset of the dataset. These models were evaluated using four commonly used metrics for inpainting and three different image sizes. The results are presented in Table 4.1.

The results demonstrate that the trained model was not able to surpass the LaMa pre-trained model, as it scored worse on all metrics. This could be attributed to a few potential causes, such as the larger batch size of 130 used in the LaMa model, which allows the model's parameters to be adjusted based on a larger set of training samples, thus providing

Table 4.1: Comparison metrics between the trained model and the LaMa pre-trained model

Image size	Model	LPIPS ↓	FID ↓	U-IDS ↑ (%)	P-IDS ↑ (%)
256x256 px	**LaMa**	**0.4363**	**1.5313**	**34.81**	**22.22**
	Trained	0.4493	4.0668	21.69	08.65
512x512 px	**LaMa**	**0.4468**	**1.0240**	**48.71**	**25.92**
	Trained	0.4571	2.4696	27.08	12.43
1024x1024 px	**LaMa**	**0.4507**	**1.1717**	**35.41**	**21.79**
	Trained	0.4583	2.4318	26.57	11.71

a more stable optimisation. Additionally, even though the dataset used has a reasonable size of 100,000 images, the dataset used to train the LaMa model was 45 times bigger. The use of a larger dataset can enhance the generalisation of the model by providing it with a wider range of variations and scenarios and improve the ability of extracting the most meaningful information from the data.

4.4 Inpainting model output examples

This section provides a subjective evaluation of the inpainting model's capabilities by displaying several output examples. These examples show the original image on the left, with the objects to be removed outlined in red, and the resulting image on the right.

4.4.1 Removal of personal items and furniture

Figure 4.3 and Figure 4.4 exemplify the removal of personal items from indoor images, such as photographs and cloth items, which gives the rooms a cleaner appearance. In both examples, the model was able to remove the selected objects without any visible mistakes.

Figure 4.5 illustrates how the proposed solution can be used to declutter a room by removing furniture from a bedroom. However, unlike the results obtained in the previous examples, the output generated is noticeably blurrier.

Figure 4.3: Model output of the removal of personal items from a bedroom picture.

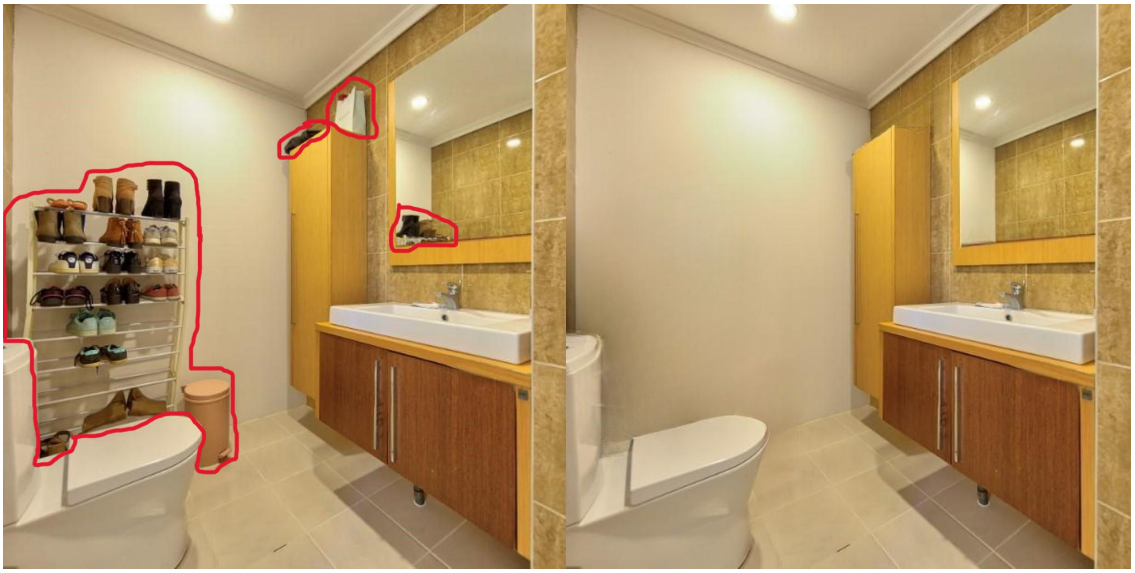

Figure 4.4: Model output of the removal of personal items from a bathroom picture.

Figure 4.5: Model output of the removal of furniture from a bedroom picture.

4.4.2 Removal of uncontrollable environmental elements

The proposed solution can also be used to remove uncontrollable environmental elements from exterior real estate images, as demonstrated in Figure 4.6 and Figure 4.7. Smaller objects such as people and utility poles were successfully removed, but the model was not able to remove larger objects such as cars, generating a blurry output.

Figure 4.6: Model output of the removal of cars from an exterior picture.

Figure 4.7: Model output of the removal of a person and a utility pole from an external picture.

4.4.3 Removal of imperfections

Another use case of the proposed solution is to remove imperfections that will be fixed before sale, as demonstrated in Figure 4.8 and Figure 4.9. In both cases, the model was able to remove the imperfections as if they had never existed. However, this use case has ethical and moral implications, as explained in subsection 3.5.2, and is not meant to deceive potential buyers by hiding defects or issues.

Figure 4.8: Model output of the removal of imperfections from an exterior picture.

Figure 4.9: Model output of the removal of construction debris from an external picture.

Chapter 5

Conclusions

This chapter concludes the project by assessing the completion of the objectives set at the beginning of the project, exploring potential future enhancements, and ending with some personal final remarks.

5.1 Summary

This project focused on the development of a deep learning inpainting solution to remove objects from real estate images, which would be beneficial to real estate agents by allowing them to improve the quality of their property images by eliminating unwanted distractions.

A comprehensive literature review was conducted to examine the evolution of inpainting models, from traditional patch-based and diffusion-based approaches to current state-of-the-art deep learning models that use complex neural networks. Four current models were studied in detail, including CoModGAN [42], which proposed a new GAN variant to have more control over the resulting images, LaMa [8], which used FFCs to use contextual information from images more efficiently, MAT [43], which used a transformer-based model with self-attention layers, and Stable Diffusion [53], which created a novel text-to-image generative model capable of removing and replacing objects from images.

This review also examined the features and drawbacks of existing real estate object removal solutions available on the market, such as Plan It All [3], which offers object removal as a paid service, and Magic Eraser [62], which provides users with a tool to remove parts of an image in real time using deep learning inpainting models. Additionally, other applications of inpainting models were discussed, including the restoration of old photographs [4, 5], self-checkout systems [7], image anomaly detection [57], and comic localisation [6].

Once the literature review was completed, the various stages of the development process were outlined. The proposed solution consists of two main components, a web application and an API to access the inpainting model. The API was created using FastAPI [65], a Python [66] web framework, and has a single endpoint that receives the original and selection mask input images and returns the output of the inpainting model. After comparing the performance of the current state-of-the-art inpainting models in a small experiment, the LaMa pre-trained model was selected for the API.

The web application was created with a React [69] front end and a .NET platform [70] back end. It enables users to upload an image and remove objects from it by using a brush tool to select the area of the objects and communicating with the API of the inpainting model. Additionally, it provides extra features for a better user experience, such as the ability to compare the original and current image, undo and redo, and alter the selection

brush colour and size. The requirements and features of the solution were analysed based on the characteristics of existing solutions and the objectives of the project. To provide real estate agents with a seamless experience, the developed solution was integrated with the Maxwork portal, the back-office portal used by RE/MAX Portugal real estate company. The solution was also made available to a select group of people and received a total of 16 reviews, with an average rating of 4.31 for ease of use and 3.90 for satisfaction with the results obtained.

Lastly, an experiment was conducted to train a new inpainting model using a dataset of real estate images previously used in sold listings. The open source code from the LaMa GitHub repository [72] was used with the same architecture as the LaMa pre-trained model. The model was trained with a batch size of 2 for 20 epochs, with the best performance achieved on the 16th epoch. The trained model was then compared to the pre-trained model using four evaluation metrics, FID [40], LPIPS [41], P-IDS [42], and U-IDS [42]. The results showed that the new trained model was not able to outperform the pre-trained model, as it scored worse on all metrics.

5.2 Fulfilled objectives

The six objectives set at the beginning of the project at section 1.3 were completed. A literature review was conducted to explore existing deep learning inpainting models and image object removal solutions for real estate. A new real estate image object removal solution was developed and incorporated into the Maxwork portal. To assess the performance and viability of this solution, it was evaluated on several aspects, namely inpainting metrics, average response time, and user feedback. An experiment was also conducted to train a new model using an open source inpainting model on a dataset of real estate images.

5.3 Limitations and future work

Hardware restrictions posed a challenge to the project, as they limited the performance of the inpainting model trained using real estate images. The graphical computational capacity restricted the potential of the trained model to achieve better results than the pre-trained model. In addition, augmentation techniques, such as image rotation and flip, could be used to artificially increase the size of the dataset, improving the model's generalisation and robustness. These factors could potentially lead to a model that performs better than the pre-trained one.

Another limiting factor of hardware restrictions is the deployment of the solution in a production environment. To ensure a good experience for the thousands of users who access the Maxwork platform daily, the solution must be deployed in a system with multiple GPUs, which increases operational costs.

The developed solution could be enhanced in the future by incorporating additional image editing capabilities, turning it into a complete image editing package. Examples of such tools that could be beneficial for real estate agents include cropping and rotating images, upscaling, enhancement filters, day-to-dusk transitions, and virtual staging.

5.4 Final remarks

The primary goal of this project was to integrate an image object removal tool into the RE/MAX Portugal back-office portal. This is of great value to real estate agents, as any tool that can boost the visibility of their ads is essential. By integrating this tool into their back-office platform, agents can use it without disrupting their current workflow.

The model's performance was not ideal when it came to eliminating large objects such as cars and beds, but it was able to successfully remove smaller objects without any visible errors. Considering the constant advancement of inpainting models in recent years, these issues are likely to be resolved in the near future.

Although this type of tool can be very useful for real estate agents, it is essential to consider the ethical and moral implications of its use. It is important to ensure that the tool is used responsibly and not to deceive or mislead potential buyers.

Bibliography

[1] Research {and} Markets ltd. *Real Estate Market Size, Share & Trends Analysis Report by Property (Residential, Commercial, Industrial, Land), by Type (Sales, Rental, Lease), by Region, and Segment Forecasts, 2022-2030*. Apr. 2022. URL: https://www.researchandmarkets.com/reports/4514489/real-estate-market-size-share-and-trends (visited on 28/11/2022).

[2] *Real Estate Decluttering Service | Unwanted Objects Removal | Phixer*. Phixer Declutter. URL: https://www.phixer.net/services/object-removal/ (visited on 03/01/2023).

[3] *Object Removal on real estate picture*. Plan It All. URL: https://www.plan-it-all.com/photo-enhancement/object-removal/ (visited on 03/01/2023).

[4] Ziyu Wan et al. 'Bringing old photos back to life'. In: *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 2747–2757.

[5] Ziyu Wan et al. 'Old photo restoration via deep latent space translation'. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022).

[6] Minshan Xie et al. 'Seamless manga inpainting with semantics awareness'. In: *ACM Transactions on Graphics* 40.4 (19th July 2021), 96:1–96:11. ISSN: 0730-0301. DOI: 10.1145/3450626.3459822. URL: https://doi.org/10.1145/3450626.3459822 (visited on 04/12/2022).

[7] Vojtěch Bartl, Jakub Špaňhel and Adam Herout. 'PersonGONE: Image Inpainting for Automated Checkout Solution'. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). ISSN: 2160-7516. June 2022, pp. 3114–3122. DOI: 10.1109/CVPRW56347.2022.00351.

[8] Roman Suvorov et al. *Resolution-robust Large Mask Inpainting with Fourier Convolutions*. 10th Nov. 2021. DOI: 10.48550/arXiv.2109.07161. arXiv: 2109.07161[cs, eess]. URL: http://arxiv.org/abs/2109.07161 (visited on 03/12/2022).

[9] Xiaobo Zhang et al. 'Image inpainting based on deep learning: A review'. In: *Information Fusion* 90 (1st Feb. 2023), pp. 74–94. ISSN: 1566-2535. DOI: 10.1016/j.inffus.2022.08.033. URL: https://www.sciencedirect.com/science/article/pii/S1566253522001324 (visited on 10/12/2022).

[10] Zhen Qin et al. 'Image inpainting based on deep learning: A review'. In: *Displays* 69 (1st Sept. 2021), p. 102028. ISSN: 0141-9382. DOI: 10.1016/j.displa.2021.102028. URL: https://www.sciencedirect.com/science/article/pii/S0141938221000391 (visited on 03/12/2022).

[11] David Josué Barrientos Rojas, Bruno José Torres Fernandes and Sergio Murilo Maciel Fernandes. 'A Review on Image Inpainting Techniques and Datasets'. In: *2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*. 2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI). ISSN: 2377-5416. Nov. 2020, pp. 240–247. DOI: 10.1109/SIBGRAPI51738.2020.00040.

[12] Jireh Jam et al. 'A comprehensive review of past and present image inpainting methods'. In: *Computer Vision and Image Understanding* 203 (1st Feb. 2021), p. 103147.

ISSN: 1077-3142. DOI: 10.1016/j.cviu.2020.103147. URL: https://www.sciencedirect.com/science/article/pii/S1077314220301661 (visited on 03/12/2022).

[13] Hanyu Xiang et al. 'Deep learning for image inpainting: A survey'. In: *Pattern Recognition* 134 (1st Feb. 2023), p. 109046. ISSN: 0031-3203. DOI: 10.1016/j.patcog.2022.109046. URL: https://www.sciencedirect.com/science/article/pii/S003132032200526X (visited on 18/08/2023).

[14] A.A. Efros and T.K. Leung. 'Texture synthesis by non-parametric sampling'. In: *Proceedings of the Seventh IEEE International Conference on Computer Vision*. Proceedings of the Seventh IEEE International Conference on Computer Vision. Vol. 2. Sept. 1999, 1033–1038 vol.2. DOI: 10.1109/ICCV.1999.790383.

[15] Marcelo Bertalmio et al. 'Image inpainting'. In: *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. SIGGRAPH '00. USA: ACM Press/Addison-Wesley Publishing Co., 1st July 2000, pp. 417–424. ISBN: 978-1-58113-208-3. DOI: 10.1145/344779.344972. URL: https://doi.org/10.1145/344779.344972 (visited on 10/12/2022).

[16] Connelly Barnes et al. 'PatchMatch: A randomized correspondence algorithm for structural image editing'. In: *ACM Trans. Graph.* 28.3 (2009), p. 24.

[17] Olivier Le Meur, Mounira Ebdelli and Christine Guillemot. 'Hierarchical Super-Resolution-Based Inpainting'. In: *IEEE Transactions on Image Processing* 22.10 (2013), pp. 3779–3790. DOI: 10.1109/TIP.2013.2261308.

[18] T Chan. 'Local inpainting models and TV inpainting'. In: *SIAM J. Appl. Math.* 62.3 (2001), pp. 1019–1043.

[19] Haodong Li, Weiqi Luo and Jiwu Huang. 'Localization of Diffusion-Based Inpainting in Digital Images'. In: *IEEE Transactions on Information Forensics and Security* 12.12 (2017), pp. 3050–3064. DOI: 10.1109/TIFS.2017.2730822.

[20] Alexandru Telea. 'An image inpainting technique based on the fast marching method'. In: *Journal of graphics tools* 9.1 (2004), pp. 23–34.

[21] Yann LeCun, Yoshua Bengio and Geoffrey Hinton. 'Deep learning'. In: *nature* 521.7553 (2015), pp. 436–444.

[22] Ian J. Goodfellow et al. *Generative Adversarial Networks*. 10th June 2014. DOI: 10.48550/arXiv.1406.2661. arXiv: 1406.2661[cs,stat]. URL: http://arxiv.org/abs/1406.2661 (visited on 28/11/2022).

[23] Deepak Pathak et al. 'Context Encoders: Feature Learning by Inpainting'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[24] Satoshi Iizuka, Edgar Simo-Serra and Hiroshi Ishikawa. 'Globally and locally consistent image completion'. In: *ACM Transactions on Graphics (ToG)* 36.4 (2017), pp. 1–14.

[25] Jiahui Yu et al. 'Generative image inpainting with contextual attention'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 5505–5514.

[26] Martin Arjovsky, Soumith Chintala and Léon Bottou. 'Wasserstein generative adversarial networks'. In: *International conference on machine learning*. PMLR. 2017, pp. 214–223.

[27] Guilin Liu et al. *Image Inpainting for Irregular Holes Using Partial Convolutions*. 15th Dec. 2018. DOI: 10.48550/arXiv.1804.07723. arXiv: 1804.07723[cs]. URL: http://arxiv.org/abs/1804.07723 (visited on 04/12/2022).

[28] Olaf Ronneberger, Philipp Fischer and Thomas Brox. 'U-net: Convolutional networks for biomedical image segmentation'. In: *International Conference on Medical image computing and computer-assisted intervention*. Springer. 2015, pp. 234–241.

[29] Kamyar Nazeri et al. *EdgeConnect: Generative Image Inpainting with Adversarial Edge Learning*. 11th Jan. 2019. DOI: 10.48550/arXiv.1901.00212. arXiv: 1901.00212[cs]. URL: http://arxiv.org/abs/1901.00212 (visited on 28/11/2022).

[30] Jiahui Yu et al. 'Free-form image inpainting with gated convolution'. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 4471–4480.

[31] Takeru Miyato et al. 'Spectral normalization for generative adversarial networks'. In: *arXiv preprint arXiv:1802.05957* (2018).

[32] Phillip Isola et al. 'Image-to-image translation with conditional adversarial networks'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1125–1134.

[33] Andrés Romero et al. 'NTIRE 2022 Image Inpainting Challenge: Report'. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). ISSN: 2160-7516. June 2022, pp. 1149–1181. DOI: 10.1109/CVPRW56347.2022.00124.

[34] Tero Karras, Samuli Laine and Timo Aila. *A Style-Based Generator Architecture for Generative Adversarial Networks*. 29th Mar. 2019. DOI: 10.48550/arXiv.1812.04948. arXiv: 1812.04948[cs,stat]. URL: http://arxiv.org/abs/1812.04948 (visited on 20/12/2022).

[35] Olga Russakovsky et al. 'Imagenet large scale visual recognition challenge'. In: *International journal of computer vision* 115.3 (2015), pp. 211–252.

[36] *WikiArt.org - Visual Art Encyclopedia*. www.wikiart.org. URL: https://www.wikiart.org/ (visited on 21/12/2022).

[37] Bolei Zhou et al. 'Places: A 10 Million Image Database for Scene Recognition'. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.6 (June 2018). Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 1452–1464. ISSN: 1939-3539. DOI: 10.1109/TPAMI.2017.2723009.

[38] Jari Korhonen and Junyong You. 'Peak signal-to-noise ratio revisited: Is simple beautiful?' In: *2012 Fourth International Workshop on Quality of Multimedia Experience*. 2012 Fourth International Workshop on Quality of Multimedia Experience. July 2012, pp. 37–38. DOI: 10.1109/QoMEX.2012.6263880.

[39] Zhou Wang et al. 'Image quality assessment: from error visibility to structural similarity'. In: *IEEE Transactions on Image Processing* 13.4 (Apr. 2004). Conference Name: IEEE Transactions on Image Processing, pp. 600–612. ISSN: 1941-0042. DOI: 10.1109/TIP.2003.819861.

[40] Martin Heusel et al. *GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium*. 12th Jan. 2018. DOI: 10.48550/arXiv.1706.08500. arXiv: 1706.08500[cs,stat]. URL: http://arxiv.org/abs/1706.08500 (visited on 27/11/2022).

[41] Richard Zhang et al. *The Unreasonable Effectiveness of Deep Features as a Perceptual Metric*. 10th Apr. 2018. DOI: 10.48550/arXiv.1801.03924. arXiv: 1801.03924[cs]. URL: http://arxiv.org/abs/1801.03924 (visited on 27/11/2022).

[42] Shengyu Zhao et al. *Large Scale Image Completion via Co-Modulated Generative Adversarial Networks*. 18th Mar. 2021. DOI: 10.48550/arXiv.2103.10428. arXiv: 2103.10428[cs]. URL: http://arxiv.org/abs/2103.10428 (visited on 03/12/2022).

[43] Wenbo Li et al. *MAT: Mask-Aware Transformer for Large Hole Image Inpainting*. 26th June 2022. DOI: 10.48550/arXiv.2203.15270. arXiv: 2203.15270[cs]. URL: http://arxiv.org/abs/2203.15270 (visited on 03/12/2022).

[44] Robin Rombach et al. *High-Resolution Image Synthesis with Latent Diffusion Models*. 13th Apr. 2022. DOI: 10.48550/arXiv.2112.10752. arXiv: 2112.10752[cs]. URL: http://arxiv.org/abs/2112.10752 (visited on 03/12/2022).

[45] Aaron Van den Oord et al. 'Conditional image generation with pixelcnn decoders'. In: *Advances in neural information processing systems* 29 (2016).

[46] Lu Chi, Borui Jiang and Yadong Mu. 'Fast fourier convolution'. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 4479–4488.

[47] Justin Johnson, Alexandre Alahi and Li Fei-Fei. 'Perceptual losses for real-time style transfer and super-resolution'. In: *European conference on computer vision*. Springer. 2016, pp. 694–711.

[48] Ziwei Liu et al. 'Deep Learning Face Attributes in the Wild'. In: Proceedings of the IEEE International Conference on Computer Vision. 2015, pp. 3730–3738. URL: https://openaccess.thecvf.com/content_iccv_2015/html/Liu_Deep_Learning_Face_ICCV_2015_paper.html (visited on 27/11/2022).

[49] Manyu Zhu et al. 'Image inpainting by end-to-end cascaded refinement with mask awareness'. In: *IEEE Transactions on Image Processing* 30 (2021), pp. 4855–4866.

[50] Prakhar Kulshreshtha, Brian Pugh and Salma Jiddi. *Feature Refinement to Improve High Resolution Image Inpainting*. 29th June 2022. DOI: 10.48550/arXiv.2206.13644. arXiv: 2206.13644[cs,eess]. URL: http://arxiv.org/abs/2206.13644 (visited on 03/12/2022).

[51] Zeyu Lu et al. *GLaMa: Joint Spatial and Frequency Loss for General Image Inpainting*. 14th May 2022. DOI: 10.48550/arXiv.2205.07162. arXiv: 2205.07162[cs]. URL: http://arxiv.org/abs/2205.07162 (visited on 03/12/2022).

[52] Ashish Vaswani et al. 'Attention is all you need'. In: *Advances in neural information processing systems* 30 (2017).

[53] Jascha Sohl-Dickstein et al. 'Deep Unsupervised Learning using Nonequilibrium Thermodynamics'. In: *Proceedings of the 32nd International Conference on Machine Learning*. International Conference on Machine Learning. ISSN: 1938-7228. PMLR, 1st June 2015, pp. 2256–2265. URL: https://proceedings.mlr.press/v37/sohl-dickstein15.html (visited on 27/08/2023).

[54] Zhisheng Xiao, Karsten Kreis and Arash Vahdat. *Tackling the Generative Learning Trilemma with Denoising Diffusion GANs*. 4th Apr. 2022. DOI: 10.48550/arXiv.2112.07804. arXiv: 2112.07804[cs,stat]. URL: http://arxiv.org/abs/2112.07804 (visited on 27/08/2023).

[55] Shaoan Xie et al. 'SmartBrush: Text and Shape Guided Object Inpainting With Diffusion Model'. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023, pp. 22428–22437. URL: https://openaccess.thecvf.com/content/CVPR2023/html/Xie_SmartBrush_Text_and_Shape_Guided_Object_Inpainting_With_Diffusion_Model_CVPR_2023_paper.html (visited on 18/08/2023).

[56] Su Wang et al. 'Imagen Editor and EditBench: Advancing and Evaluating Text-Guided Image Inpainting'. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023, pp. 18359–18369. URL: https://openaccess.thecvf.com/content/CVPR2023/html/Wang_Imagen_Editor_and_EditBench_Advancing_and_Evaluating_Text-Guided_Image_Inpainting_CVPR_2023_paper.html (visited on 18/08/2023).

[57] Jonathan Pirnay and Keng Chai. *Inpainting Transformer for Anomaly Detection*. 26th Nov. 2021. DOI: 10.48550/arXiv.2104.13897. arXiv: 2104.13897[cs]. URL: http://arxiv.org/abs/2104.13897 (visited on 04/12/2022).
[58] Diederik P Kingma and Max Welling. 'Auto-encoding variational bayes'. In: *arXiv preprint arXiv:1312.6114* (2013).
[59] *Virtual Decluttering & Object Removal at $8 per image | Styldod*. Styldod. URL: https://www.styldod.com/object-removal (visited on 03/01/2023).
[60] *Removing Object From Photo | Item Removal Photo Editing Service - revivoto*. Revivoto. URL: https://revivoto.com/services/item-removal/ (visited on 03/01/2023).
[61] *Cleanup.pictures - Remove objects, people, text and defects from any picture for free*. Cleanup Pictures. URL: https://cleanup.pictures (visited on 03/01/2023).
[62] *Magic Eraser : Remove unwanted things in seconds*. Magic Eraser. URL: https://magicstudio.com/magiceraser (visited on 03/01/2023).
[63] *Official Adobe Photoshop | Photo and design software*. Adobe Photoshop. URL: https://www.adobe.com/products/photoshop.html (visited on 03/01/2023).
[64] *Remove Unwanted Objects & Fix Imperfections with Inpaint Online!* Inpaint. URL: https://theinpaint.com/ (visited on 03/01/2023).
[65] *FastAPI documentation*. URL: https://fastapi.tiangolo.com/ (visited on 10/08/2023).
[66] Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009. ISBN: 1441412697.
[67] *TorchScript — PyTorch 2.0 documentation*. URL: https://pytorch.org/docs/stable/jit.html (visited on 10/08/2023).
[68] Adam Paszke et al. 'PyTorch: An Imperative Style, High-Performance Deep Learning Library'. In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper_files/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html (visited on 17/08/2023).
[69] *React documentation*. URL: https://react.dev/ (visited on 10/08/2023).
[70] *.NET documentation*. URL: https://learn.microsoft.com/en-us/dotnet/ (visited on 10/08/2023).
[71] Robert C Martin. *Clean architecture*. 2017.
[72] Roman Suvorov et al. *advimman/lama: LaMa Image Inpainting, Resolution-robust Large Mask Inpainting with Fourier Convolutions, WACV 2022*. URL: https://github.com/advimman/lama (visited on 15/08/2023).

Appendix A. Trained model configuration file

```
training_model:
  kind: default
  visualize_each_iters: 1000
  concat_mask: true
  store_discr_outputs_for_vis: true
losses:
  l1:
    weight_missing: 0
    weight_known: 10
  perceptual:
    weight: 0
  adversarial:
    kind: r1
    weight: 10
    gp_coef: 0.001
    mask_as_fake_target: true
    allow_scale_mask: true
  feature_matching:
    weight: 100
  resnet_pl:
    weight: 30
    weights_path: ${env:TORCH_HOME}
generator:
  kind: ffc_resnet
  input_nc: 4
  output_nc: 3
  ngf: 64
  n_downsampling: 3
  n_blocks: 18
  add_out_act: sigmoid
  init_conv_kwargs:
    ratio_gin: 0
    ratio_gout: 0
    enable_lfu: false
  downsample_conv_kwargs:
    ratio_gin: ${generator.init_conv_kwargs.ratio_gout}
    ratio_gout: ${generator.downsample_conv_kwargs.ratio_gin}
    enable_lfu: false
  resnet_conv_kwargs:
    ratio_gin: 0.75
    ratio_gout: ${generator.resnet_conv_kwargs.ratio_gin}
    enable_lfu: false
data:
  batch_size: 2
  val_batch_size: 2
  num_workers: 3
  train:
```

```
    indir: ${location.data_root_dir}/train
    out_size: 256
    mask_gen_kwargs:
      irregular_proba: 1
      irregular_kwargs:
        max_angle: 4
        max_len: 200
        max_width: 100
        max_times: 5
        min_times: 1
      box_proba: 1
      box_kwargs:
        margin: 10
        bbox_min_size: 30
        bbox_max_size: 150
        max_times: 4
        min_times: 1
      segm_proba: 0
    transform_variant: distortions
    dataloader_kwargs:
      batch_size: ${data.batch_size}
      shuffle: true
      num_workers: ${data.num_workers}
  val:
    indir: ${location.data_root_dir}/val
    img_suffix: .jpg
    dataloader_kwargs:
      batch_size: ${data.val_batch_size}
      shuffle: false
      num_workers: ${data.num_workers}
location:
  data_root_dir: /dataset
  out_root_dir: /experiments
  tb_dir: /tb_logs
  pretrained_models: /models
discriminator:
  kind: pix2pixhd_nlayer
  input_nc: 3
  ndf: 64
  n_layers: 4
optimizers:
  generator:
    kind: adam
    lr: 0.001
  discriminator:
    kind: adam
    lr: 0.0001
visualizer:
  kind: directory
  outdir: /experiments
  key_order:
  - image
  - predicted_image
  - discr_output_fake
  - discr_output_real
  - inpainted
  rescale_keys:
  - discr_output_fake
  - discr_output_real
```

```
evaluator:
  kind: default
  inpainted_key: inpainted
  integral_kind: ssim_fid100_f1
trainer:
  kwargs:
    gpus: -1
    max_epochs: 20
    gradient_clip_val: 1
    log_gpu_memory: None
    limit_train_batches: 20000
    val_check_interval: ${trainer.kwargs.limit_train_batches}
    log_every_n_steps: 1000
    precision: 32
    terminate_on_nan: false
    check_val_every_n_epoch: 1
    num_sanity_val_steps: 8
    limit_val_batches: 1000
    replace_sampler_ddp: false
  checkpoint_kwargs:
    verbose: true
    save_top_k: 5
    save_last: true
    period: 1
    monitor: val_ssim_fid100_f1_total_mean
    mode: max
```

Listing A.1: Trained model configuration file