*Escribe aquí*
*tu frase favorita.*

E indica aquí su autor

# Agradecimientos

Me gustaría agradecer...

También quiero destacar...

Por último...

# Índice general

# Índice de figuras

# Índice de tablas

# Índice de algoritmos

# Abstract

# 1 Introduction

Escribe aquí la introducción de tu Trabajo Fin de Máster, utilizando tantas secciones, subsecciones y subsubsecciones como estimes necesarias.

## 1.1. Context

**Esta palabra** está en negrita. *Esta palabra* está en cursiva. **Esta palabra** se destaca en púrpura.

## 1.2. Objectives

En la sección **??** se muestran ejemplos de palabras en negrita, cursiva y destacadas en púrpura.

Una Red Generativa Antagónica o *Generative Adversarial Network* (GAN) es... (Goodfellow et al., 2014).

Goodfellow et al. (2014) diseñaron las redes generativas antagónicas como...

### 1.2.1. Main objective

### 1.2.2. Specific objectives

1. **Objetivo parcial 1.**

2. **Objetivo parcial 2.**

3. **Objetivo parcial 3.**

# Theoretical framework 2

## 2.1. Problem Definition

The problem of image registration is defined as... (insert equation)

## 2.2. Displacement Fields

## 2.3. Diffeomorphism

## 2.4. Classical Methods

Demons, other iterative methods...

## 2.5. Deep learning-based methods

When approaching the original problem of image registration... When using deep learning, we don't directly calculate the

Differences between unsupervised and supervised deep learning methods.

### 2.5.1. U-Net

Some paper of implementation using U-Net or similar model

## 2.6. Transformers

Transformers were introduced ...

Currently, state of the art in sequence processing, tasks such as NLP...

### 2.6.1. Mechanism of Self-attention

When using transformers, the input sequence is tokenized, obtaining a set of tokens - a collection of distinct, unordeded elements.

The next step is the projection of the tokens into a distributed geometrical space of continuous-valued vectors - what is referred to as an embedding. This is done in order to preserve the semantical relationships among the tokens. In this low-dimensional space, we expect that the projections of the original tokens which hold stronger semantic relationships with each other are, indeed, closer to each other than their counterparts.

However, after the tokenization step, we have effectively lost the order of the original sequence. In order to maintain the notion of order necessary to process the input as a sequence, transformer blocks use positional encoding.

Positional encoding alters the embeddings depending of the position of the token in the original sequence.

One of the possible techniques for implementing positional encoding is using the sinousoidal function.

We then perform feature-based attention on the resulting embeddings. The inductive bias by which feature-based attention is based upon, is to allow the network to place its attention not only based in the original order of the sequence, but taking into account the content of the tokens.

In order to quantify the similarity of tokens, we will calculate the vector similarity among the tokens' projections into the learned embedding space. As this is a low-dimensional space, we can make use of the inner vector product.

The transformer uses 3 different representations of the embedding matrix, the queries, keys and values.

For each token (i.e. vector), we create a query vector, key vector and value vector. The query, key and value matrices are learned as part of the training process. In order to reduce the computational complexity of the operation, this resulting vectors are usually in lower-dimensional spaces.

The concepts of key, value and query are originally part of information retrieval systems.

Self attention is performed for each position.

The main idea is that we compute the similarity of each token of the sequence with each other token. Then, the result of the self-attention is a vector, result of the sum of each token in the sequence multiplied by its similarity score. In this way, we are capturing the most important context individually for each of the tokens in the sequence.

Self-attention with matrices First, we obtain the query, key and value matrices by multiplying the embedding matrix with the learned weight matrices WQ, WK, WV. Each of the rows of the embedding matrix corresponds to a token of the input sequence.

$$softmax(\frac{QxK^T}{\sqrt{d_k}})V = Z$$

The softmax outputs a probability distribution, with all of its components adding to one.

This concept of self-attention is further developed with multi-headed attention.

### 2.6.2. Multi-head Self-attention

With multi-headed attention, we repeat the original self-attention process, obtaining multiple representational spaces, in the form of multiple sets of query, key and value matrices. Theoretically, this allows the model to perform attention in different, independent low-dimensional spaces, which translates to the ability to jointly perform attention from different representation subspaces, at different positions.

## 2.7. Visual transformers

# Methodology

3

## 3.1. Proposed Architecture

## 3.2. Image Similarity Metric

# Experiments

4

# Conclusion and Limitations

5

## 5.1. Conclusion

## 5.2. Limitations

# Annex A

A

# Bibliografía

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., y Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.