UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA
Denkleiers • Leading Minds • Dikgopolo tša Dihlalefi

# COS700 Research Proposal

# Vision Transformers for Pest and Disease Detection

Andreas Louw
u15048366

u15048366@tuks.co.za

**Supervisor(s)**:
Dr. Anna Bosman
Mr. Arné Schreuder

**July 2023**

# Vision Transformers for Pest and Disease Detection


## Abstract

The ability for researchers and industry to identify and act upon observation of pests and diseases in the forestry and agriculture sector is important for the bio-security of a country. A reliable mechanism by which pest and diseases can be identified is crucial to ensure biosecurity. The image-processing research field has improved significantly in the last decade. Traditional approaches make use of *deep neural network*s (DNNs), in particular *convolutional neural network*s (CNNs). Much research has been done to extend on this field [1]. This includes pre-training, transfer learning, few-shot approaches, transformers and diffusion techniques. *Vision Transformer*s (ViTs) pose a potential further improvement to the image classification problem for pests and diseases, especially where the visual similarities between species are fine-grained [2]. This research investigates different image classification techniques and architectures, comparing traditional CNNs to ViTs on a number of pest and disease datasets. All of the approaches make use of transfer learning from pre-trained, backbone networks. An empirical analysis is done to determine the effects of network size on the performance of these networks.



## Keywords:

artificial neural networks, deep neural networks, convolutional neural networks, transformers, vision transformers

# 1 Introduction

Domestic agriculture is important for any country to ensure food and material security for the populace. Pests and diseases are a threat to the security of food and material production. Work in the plant science field usually requires physical samples to be taken in the field. As pests and diseases can vary in types it is important to state that a full diagnoses cannot be made solely on images. Classifying pests and diseases with the help of image recognition is only a tool to help diagnoses. Depending on the situation further scientific methods should be followed to fully and reliably diagnose the pest and/or disease. This research is interested in comparing *convolutional neural network*s (CNNs) [1] with *Vision Transformer*s (ViTs) [2] when the goal is to detect pests and diseases on plants.

*Artificial neural network*s (ANNs) have been used in various ways to solve problems and speed up solving complex problems [3]. One of the main subfields of ANNs is deep learning [3]. One of the applications of DNNs is image recognition. Image recognition in artificial intelligence is the identifying and categorising of objects and/or patterns in an image or video. The current state of the art for image recognition tasks with deep learning is CNNs [1]. The neural network uses a convolutional layer that recognises patterns and features in the given image or video. A convolution is a mathematical operation that is preformed on two functions producing a third function. The third function expresses how the shape of the one function affects the shape of the other function.

An alternative to image recognition with CNN are ViT, ViTs are a deep learning model architecture that uses the transformer architecture for image recognition tasks [2]. The transformers neural network architecture was first introduced in the paper [4]. The transformers architecture was made popular by the *bidirectional encoder representations from transformers* (BERT) model used for natural language processing [5] whereafter it was adapted to image recognition tasks introduced in the paper [2].

The rest of the paper is set out as follows, the problem statement follows in Section 2 that elaborates on the motivation of this research. Section 3 contains the literature study conducted on how this problem is currently solved and investigates past and current research of ViTs. Section 4 sets out the methodology followed for this research. Section 5 outlines the planning for this research.

# 2 Problem Statement

The aim of this research is to discover whether ViT is a better approach than current CNN to train AI models with supervised learning on a dataset for pest and disease detection. The research aims at investigating the relevance of using ViT for early pest and disease detection in plant sciences. The main questions to be answered by this research are:

- How do ViTs compare to CNNs when dealing with pest and disease detection with images? Is the accuracy better or worse when both CNN and ViT is trained with the same dataset?
- Is ViT model more computational efficient than the CNN model?
  - Does ViT converge faster than CNN when training?
  - Does ViT use less memory than CNN when training?
- Investigate the effects of image resolution and scaling when training ViT.

# 3 Literature Study

The following section provides background for the research. The subsections that follows are: Section 3.1 that gives background on ANNs the building block of *artificial intelligence* (AI), with section 3.2 that gives background on *feed-forward neural network* (FFNN) and section 3.3 on training of neural networks. Section 3.4 introduces backpropagation. Section 3.5 covers CNNs that is the current state of the art for image processing. Section 3.6 discusses transformers that is the current state of the art for language models. Section 3.7 coversViTs that is a new architecture used for image processing, and Section 3.8 covers related work.

## 3.1 Artificial Neural Networks

This subsection introduces ANNs. ANNs were inspired by the function and structure of biological neural networks like the human brain [3]. This biological neural network consists of interconnected neurons.

The artificial neuron, illustrated in Figure 1, is the building block of ANNs. The artificial neuron is a mathematical model that simulates a biological neuron. The artificial neuron receives one or multiple input values,

gets the weighted sum of the inputs and passes the result through an activation function $f(x)$ along with the bias $b$ for the neuron that produces an output as shown in equation (1).

$$y = f(\Sigma xW + b) \tag{1}$$

Figure 1 illustrates how a artificial neuron works. Inputs $X_n$ along with a weight $W_n$ are summised. The summised total along with the bias is then used in an activation function, the activation function is the mechanism that decides if the neuron is activated or not. The output of the activation function is the output for the neuron.

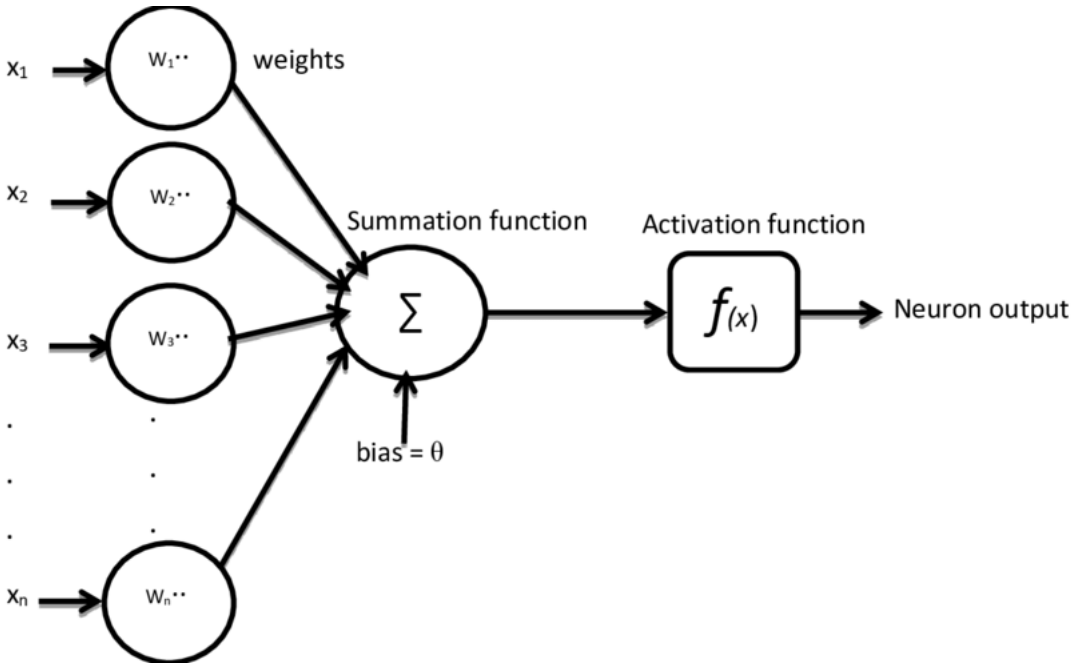


Figure 1: Artificial neuron illustration as taken from [6]. The figure illustrates how the summation (centre) of the inputs (left hand side) with weight then passed to the activation function (right hand side) to create the output.

The bias is a constant value added to the input of each neuron [7].

Activation functions are mathematical functions that add non-linearity to the output of neurons. The activation function determines if a neuron should be activated or not, based on the weighted sum of its inputs. Activation functions enable modelling of complex relationships in data The vanishing gradient problem is the phenomenon where the gradients of the loss functions become small. As a result convergence is slow and training performance is poor [8].

One of the most commonly used activation functions is the *Rectified linear unit* (ReLU) [9]. The ReLU formula is as shown in Eq (2).

$$f(x) = max\{0, x\} \tag{2}$$

The last major component of an ANN is the error function. An error function can also be referred to as loss function, cost function or objective function. An error function is a mathematical function that quantifies the error between the predicted output and the desired output. The error function is used to measure the performance of a model.

## 3.2 Feedforward Neural Network

This section introduces the necessary background for FFNN. FFNN is a neural network that is made of multiple layers with interconnected nodes. Information flows in a unidirectional manner form input to output layer. Each neuron calculates the weighted sum of inputs and applies the activation function this produces the output [10] as shown in Figure 2.

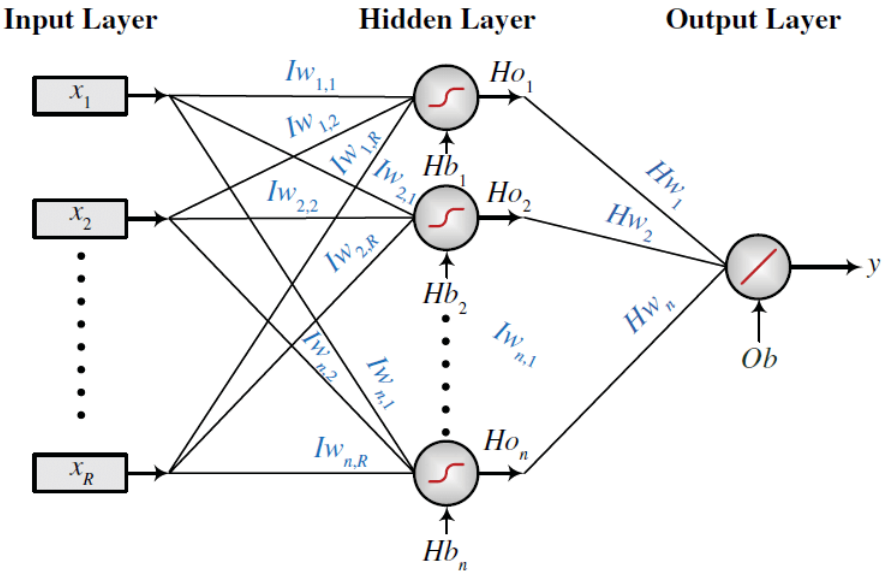


Figure 2: FFNN Model Architecture as taken from [11]. The input layers (left hand side), feed into the hidden layers (centre) with weights, in the hidden layer the activation function is applied to the input along with the weights, this produces the output in the output layer (right hand side).

## 3.3 Training

This subsection introduces the necessary background for Training of a neural network [12]. Training a neural network is the process of optimising the weights and biases of the model to learn the mapping between the inputs and outputs. During supervised training the weights and biases are adjusted iteratively based on the difference between the output and the desired outputs or actual values. The process of adjusting the weights and biases to minimise the difference between the output and desired output is known as backpropagation or optimisation. The goal of training a neural network is to minimise the error to allow the model to adjust well with new unseen data. There are several types of training; supervised learning, unsupervised learning, semi-supervised learning, reinforcement learning and transfer learning. Supervised learning uses datasets where the data is labelled, input-output pairs are used to train the model. The model learns to map the inputs to the outputs based on the given labels [13].

## 3.4 Backpropagation

This subsection introduces the necessary background on backpropagation. Backpropagation [14] is typically used when training CNN, it is a gradient-based optimisation algorithm that updates the weights of the network during training. Backpropagation calculates the gradient of the loss function with respect to the weights of the network and uses the gradient to update the weights so that the loss function is minimised.

## 3.5 Convolutional Neural Networks

This section introduces necessary background on CNNs. A CNNs are a type of deep learning model that is used for processing videos and images. CNNs are designed to extract and learn features from inputs such as images or videos. Currently CNNs is state of the art performance in image classification, object detection and image generation. CNNs consist out of the following concepts: convolutional layers, pooling layers, fully connected layers and backpropagation as illustrated in Figure 3. The following paragraph expands on these concepts.

Convolutional layers [15] applies convolutional operations on a given input. The convolutional operations that are preformed on the inputs include: computing dot products between filter and local input patch at each position, sliding small filters over the input image. These operations allow the

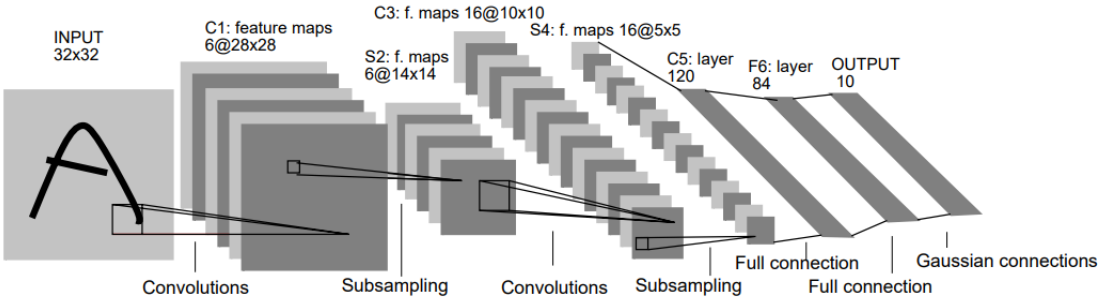


Figure 3: CNN model architecture as taken from [15]. An image is taken as input (left hand side), where the input is used in the convolutional layer (labelled as C for example C1: feature map) thereafter the out put of the convolutional layer is used in the sub-sampling layer (labelled as S for example S2 f. maps), the convolutional layer along with the sub-sampling layer is seen as one layer. The output of these layers are then used to make predictions.

network to learn features like edges, textures and patterns form the input. Pooling layers [16] are usually included in CNNs where the pooling layers downsample the feature maps created by the convolutional layers. Pooling aids in reducing the spatial dimensions of the feature maps created, this reduce the computational cost of the network and allows capturing of more abstract features. Fully connected layers [1] are traditional ANN layers that uses extracted features from the convolutional and pooling layers to make a final prediction. These predictions can be an image classification or an object detection.

## 3.6 Transformers

This subsection introduces the necessary background for transformers [4]. Transformers are a type of artificial deep learning model that is used in *natural language processing* (NLP) tasks. Transformers architecture relies on a self-attention mechanism that allows processing of input sequence in parallel in contrast to *recurrent neural network* (RNN) that process input sequentially. The self-attention mechanism is based of the concept of scaled dot-product attention [4]. The self-attention mechanism allows different weights to be assigned to different words in an input sequence, based on their relevance in the current context. The weights are used to calculate the weighted sum of the input words, this forms the basis of the models representations

and subsequent predictions. The use of self-attention allows transformers to capture long-range dependencies in sequence [4]. By adding to different parts of the input sequence with different levels of attention, it is possible to capture both local and global contextual information allowing modelling of complex patterns. Transformers use positional encoding techniques to encode the order of words since the word order does not have to be inherited like with RNNs. Positional encoding is typically achieved by adding fixed functions to the input embedding, to maintain the sequential order of input words during processing.

Most noticeable models using transformers are: BERT [5], *Generative Pre-trained Transformer* (GPT) [17] and *Text-to-Text Transfer Transformer* (T5) [18].

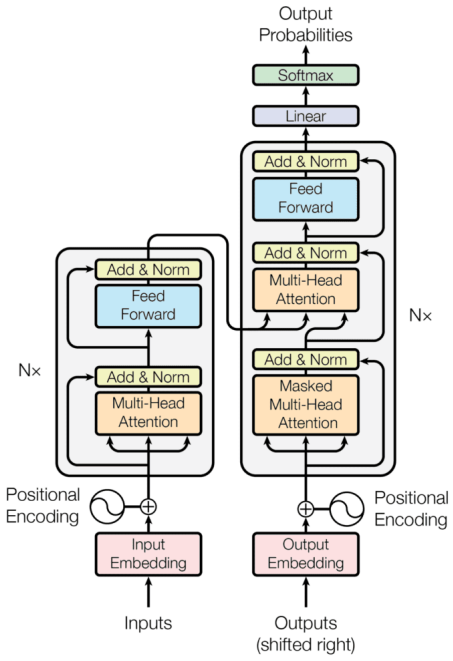


Figure 4: Transformer model architecture as taken from [4]. Text is given as input, the input is encoded, the encoded input is then assigned positional information that is then used as the input for encoder layer. The encoder (left hand side) layer maps all inputs to a abstract continuous representation that hold the learn information for the sequence. The decoder (right hand side) generates text as an output sequences by taking in the output from the encoding layer as well as the output generated by the decoder layer.

Figure 4 illustrates the Transformer model architecture, with the encoder on the left and decoder on the right. The received inputs are encoded to be

represented as a vector to capture the meaning. There after the embedded input goes through positional encoding, this allows the model to be aware of positional importance, that is needed in tasks like languages. Nx in the illustration are modules that can be stacked multiple times. The encoder consists out of mainly two layers: self-attention layer and the FFNN layer. The self-attention layer calculates the importance of a word by comparing it to other words in the input, attention scores are assigned based on importance. The output form the self-attention layer is fed into a FFNN that applies non-linear transformation to each position, allowing the model to learn complex relationships. The decoder also has both a self-attention layer and FFNN in addition to an encoder-decoder attention layer, this layer allows the model to creates output by taking the context into consideration. While training the model masking is used to stop the model form peeking at upcoming sequence, only allowing the model to have access to the previously generated output. The decoder crates output, one word at a time. At each generation the probability of the next word is predicted and the word with the highest probability is used as output, the predicted word is then used as input for predicting the next word.

## 3.7 Vision Transformers

This subsection introduces the necessary background for ViTs [2]. ViTs uses the Transformer architecture. ViT takes a new approach to image recognition tasks by breaking the image into small patches and treating each patch as a token. The patches are fed into a transformer encoder that processes it and gives a final output. ViT have been shown to have greater flexibility and scalability compared to CNNs.

ViTs consists out of the following concepts; Image Patching, Positional Embeddings, Transformer Encoder, Classification Head as illustrated in Figure 5.

The input image is divided onto patches that do not overlap, with a set fixed size, usually 16x16 or 32x32 pixels. Each patch is treated as a separate token and embedded into a one dimensional sequence.

Positional embedding encode the spatial information of the image patches and capturing their relative positions.

The patch tokens and positional embedding are processed by a series of transformer encoder blocks, that preform self-attention operations to capture local and global contextual information over the image.

The output of the transformer encoder is passed through a *multi-layer perceptron* (MLP) for final classification that predicts the labels for the image.

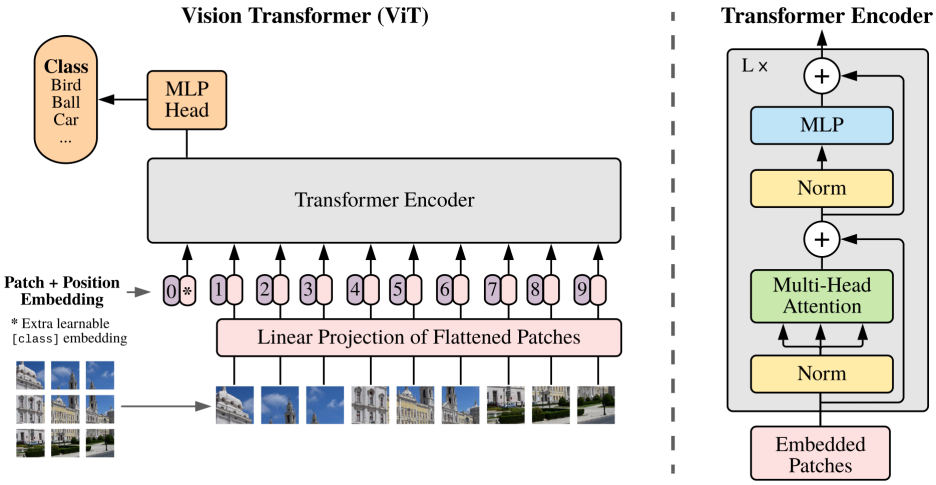


Figure 5: Vision transformer model architecture as taken from [2]. An image is given as input, the input is divided into patches (left hand side) and then flattened. positional encoding is then done on the linear projection of the flattened patches, this is then fed into the encoder (right hand side) that will produce an output.

## 3.8 Related Work

This section introduces related work in the field of pest and disease detection, highlighting the methods currently being used and some research that is done with pest and disease detection with AI. Currently there are several methods and technologies being used for pest and disease detection as provided by Martinelli et al.[19] and Li et al.[20] that provide a review of current methods and technologies.

Visual inspection is the process of trained personnel inspecting the plant for known pests and diseases and their symptoms [21]. Molecular Techniques involves techniques like *polymerase chain reaction* (PRC) and *Deoxyribonucleic acid* (DNA) sequencing [22]. These techniques involve analysing plant samples for presence of specific DNA sequences. Remote Sensing involves using satellite and drone imagery to detect changes in plant health, like changes in reflectance or thermal patterns [23]. Automated Monitoring Systems involves sensor networks and *Internet of Things* (IoT) devices to continuously monitor environment and plant health indicators [24]. Indicators that can be monitored are weather data like temperature, humidity, wind, ect.

Artificial intelligence, specifically deep learning is being used more commonly. Research done by Sladojvic et al. [25] explored plant disease detection with CNNs that is able to classify healthy and sick leaves, where it is able to

classify between 13 different diseases.

# 4 Methodology

This section introduces the methodology that the research will follow. Section 4.1 expands on the empirical process, Section 4.2 gives an overview of the datasets to be used, Section 4.3 gives a overview on the model configuration, Section 4.4 expand on the experiments to be conducted and Section 4.5 expands on how the statistical analysis will be handled for the research.

## 4.1 Empirical Process

An empirical process will be followed for the research as multiple experiments will be conducted to reject or verify the hypothesis about ViT and CNN. The experiments are as follows.

- Train a base CNN and ViT on the *Modified National Institute of Standards and Technology database* (MNIST), compare the outcomes of these models.
- Train a CNN and ViT on the pest and diseases datasets, compare the outcomes of these models.
- Compare the training time and performance of the models.
- Parameter tuning on the models to compare the effects.

## 4.2 Dataset

This section provides detail on the datasets used. Two datasets are used namely a *Forestry and Agricultural Biotechnology Institute* (FABI) dataset and the IP102 dataset. The IP102 dataset consists out of 75222 images that are labelled. An example image is given in Figure 6. These labels consist of both pests and diseases on 8 main classes of crop namely: Rice, Corn, Wheat, Beet, Alfalfa, Vitis, Citrus and Mango. The FABI dataset consists out of 2354 images, the image given in 7 gives an example. This dataset is specific to some of the pests and disease that are commonly found by FABI researchers.

Figure 6: Image taken from the IP102 dataset, presents a neoplocaederus

Figure 7: Image taken from the FABI dataset, presents shell lerp psyllid.

| **Dataset** | **Types** | **Attributes** | **Classes** | Resolution |
|---|---|---|---|---|
| MNIST | images | 70000 | 10 | 28x28 |
| IP102 | images | 75222 | 102 | 800x600 |
| FABI | images | 2354 | 8 | 1024x768 |

Table 1: Datasets

## 4.3 Models

The following subsection elaborates on the models for this research. There will be two main models that will be compared namely CNN and ViT. The configurations for the models are given in Table 2 and Table 3

The initial CNN configuration that will be used is given in Table 2 as defined in the original paper [26]:

| Parameter | Value |
|---|---|
| number of channels | 3 |
| embedding size | 64 |
| hidden sizes | [256, 512, 1024, 2048] |
| depths | [3, 4, 6, 3] |
| layer type | bottleneck |
| hidden activation function | 'relu' |
| downsample in fist stage | false |
| output features range | none |
| output indices | none |

Table 2: Table of Initial CNN Parameters.

The initial ViT configuration that will be used is given in Table 3 as defined in the original paper [2].

Both the IP102 and FABI datasets will be used for training and testing, the datasets will be split into 80% training and 20% testing.

## 4.4 Experiments

This subsection will give information on the experiments that will be conducted for this research. For each of these two models two experiments will be conducted, firstly a pre-trained model of each will be used and compared will both the IP102 and FABI datasets. The second will be training the models from scratch, where the IP102 dataset will be used to build a general model and then the FABI dataset used to specialise the model for FABI pest and disease detection. Experiments will consists of 10 runs for each model

| Parameter | Value |
|---|---|
| hidden size | 768 |
| number of hidden layers | 12 |
| number of attention heads | 12 |
| intermediate size | 3072 |
| hidden activation function | 'relu' |
| hidden dropout probability | 0.0 |
| attention dropout probability | 0.0 |
| initialiser range | 0.02 |
| layer normalisation epsilon | 1e-12 |
| image size | 224 |
| patch size | 16 |
| number channels | 3 |
| query key values bias | True |
| encoder stride | 16 |

Table 3: Table of Initial ViT Parameters.

and with each run the random seed will be changed. The run will run until completion with a maximum of 10 epochs as the stopping condition.

The expected output of each model on each run will be the average loss function for the run over all the iterations for that run, along with the run time, iteration number and step number.

The models performance will be determined as follows. The loss function used in the research is the cross entropy error. The classification accuracy will be measured at the end of each run on the testing data.

## 4.5 Statistical Analysis

A statistical analysis will be followed for analysing the experiments. Both trained models for CNN and ViT will given unseen testing data.

The outputs of the experiments will be taken then a comparison will

be made between the CNN and ViT models, this will be plotted as a graph. Possible statistical tests that will be used are *Analysis of Variance* (ANOVA) [27] statistical test, Mann-Whitney U test [28] and T-test [29], to determine statistical significance.

Each experiment will have 10 runs, a run will consists of unseen test data given as input to the trained model. Each run will have a maximum epoch stopping condition of 10.

Each run will produce a measurement of average accuracy for the run, the average loss over all the runs will give the average loss for the experiment.

# 5 Planning

This section highlights the research plan. Figure 8 outlines the timeline for the tasks. The main task to be done are: General research, writing of proposal, data collection and understanding, experiments and writing of final report.

| Task | Genral research | | | Write Proposal | Data | | | | Experiments | | | | | | Final Report Write up |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Find 4 papers on topic | Basic research on technology | | | Data Gathering | Data Understanding | Data Preperation | | Base CNN model | Base ViT model | Extend CNN with pest and disease | Extend ViT with pest and disease | Parameter tuning | |
| **Weeks** | **4** | 1 | 3 | **4** | **3** | 1 | 1 | 1 | **13** | 1 | 2 | 2 | 3 | 5 | **7** |
| 03/04/2023 | | | | | | | | | | | | | | | |
| 10/04/2023 | | | | | | | | | | | | | | | |
| 17/04/2023 | | | | | | | | | | | | | | | |
| 24/04/2023 | | | | | | | | | | | | | | | |
| 01/05/2023 | | | | | | | | | | | | | | | |
| 08/05/2023 | | | | | | | | | | | | | | | |
| 15/05/2023 | | | | | | | | | | | | | | | |
| 22/05/2023 | | | | | | | | | | | | | | | |
| 29/05/2023 | | | | | | | | | | | | | | | |
| 05/06/2023 | | | | | | | | | | | | | | | |
| 12/06/2023 | | | | | | | | | | | | | | | |
| 19/05/2023 | | | | | | | | | | | | | | | |
| 26/06/2023 | | | | | | | | | | | | | | | |
| 03/07/2023 | | | | | | | | | | | | | | | |
| 10/07/2023 | | | | | | | | | | | | | | | |
| 17/07/2023 | | | | | | | | | | | | | | | |
| 24/07/2023 | | | | | | | | | | | | | | | |
| 31/07/2023 | | | | | | | | | | | | | | | |
| 07/08/2023 | | | | | | | | | | | | | | | |
| 14/08/2023 | | | | | | | | | | | | | | | |
| 21/08/2023 | | | | | | | | | | | | | | | |
| 28/08/2023 | | | | | | | | | | | | | | | |
| 04/09/2023 | | | | | | | | | | | | | | | |
| 11/09/2023 | | | | | | | | | | | | | | | |
| 18/09/2023 | | | | | | | | | | | | | | | |
| 25/09/2023 | | | | | | | | | | | | | | | |
| 02/10/2023 | | | | | | | | | | | | | | | |
| 09/10/2023 | | | | | | | | | | | | | | | |
| 16/10/2023 | | | | | | | | | | | | | | | |
| 23/10/2023 | | | | | | | | | | | | | | | |
| 30/10/2023 | | | | | | | | | | | | | | | |

Figure 8: Planning Table

# References


[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[2] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[3] A. K. Jain, J. Mao, and K. M. Mohiuddin, "Artificial neural networks: A tutorial," *Computer*, vol. 29, no. 3, pp. 31–44, 1996.

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[6] J. A. Yacim and D. G. B. Boshoff, "Impact of artificial neural networks training algorithms on accurate prediction of property values," *Journal of Real Estate Research*, vol. 40, no. 3, pp. 375–418, 2018.

[7] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction*, vol. 2. Springer, 2009.

[8] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 6, no. 02, pp. 107–116, 1998.

[9] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814, 2010.

[10] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

[11] S. Razavi and B. A. Tolson, "A new formulation for feedforward neural networks," *IEEE Transactions on neural networks*, vol. 22, no. 10, pp. 1588–1598, 2011.

[12] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[13] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image segmentation using deep learning: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 7, pp. 3523–3542, 2021.

[14] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[15] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[16] P. Sermanet and Y. LeCun, "Traffic sign recognition with multi-scale convolutional networks," in *The 2011 international joint conference on neural networks*, pp. 2809–2813, IEEE, 2011.

[17] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[18] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 5485–5551, 2020.

[19] F. Martinelli, R. Scalenghe, S. Davino, S. Panno, G. Scuderi, P. Ruisi, P. Villa, D. Stroppiana, M. Boschetti, L. R. Goulart, *et al.*, "Advanced methods of plant disease detection. a review," *Agronomy for Sustainable Development*, vol. 35, pp. 1–25, 2015.

[20] L. Li, S. Zhang, and B. Wang, "Plant disease detection and classification by deep learning—a review," *IEEE Access*, vol. 9, pp. 56683–56698, 2021.

[21] R. Balodi, S. Bisht, A. Ghatak, K. Rao, *et al.*, "Plant disease diagnosis: technological advancements and challenges," *Indian Phytopathology*, vol. 70, no. 3, pp. 275–281, 2017.

[22] M. M. López, P. Llop, A. Olmos, E. Marco-Noales, M. Cambra, and E. Bertolini, "Are molecular tools solving the challenges posed by detection of plant pathogenic bacteria and viruses?," *Current issues in molecular biology*, vol. 11, no. 1, pp. 13–46, 2009.

[23] F. Baret, V. Houlès, and M. Guerif, "Quantification of plant stress using remote sensing observations and crop models: the case of nitrogen management," *Journal of experimental botany*, vol. 58, no. 4, pp. 869–880, 2007.

[24] I. Buja, E. Sabella, A. G. Monteduro, M. S. Chiriacò, L. De Bellis, A. Luvisi, and G. Maruccio, "Advances in plant disease detection and monitoring: From traditional assays to in-field diagnostics," *Sensors*, vol. 21, no. 6, p. 2129, 2021.

[25] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, "Deep neural networks based recognition of plant diseases by leaf image classification," *Computational intelligence and neuroscience*, vol. 2016, 2016.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[27] T. K. Kim, "Understanding one-way anova using conceptual figures," *Korean journal of anesthesiology*, vol. 70, no. 1, pp. 22–26, 2017.

[28] H. B. Mann and D. R. Whitney, "On a test of whether one of two random variables is stochastically larger than the other," *The annals of mathematical statistics*, pp. 50–60, 1947.

[29] Student, "The probable error of a mean," *Biometrika*, vol. 6, no. 1, pp. 1–25, 1908.

# A Appendix

## Acronyms

**AI** *artificial intelligence*

**ANN** *artificial neural network*

**ANOVA** *Analysis of Variance*

**BERT** *bidirectional encoder representations from transformers*

**CNN** *convolutional neural network*

**DNA** *Deoxyribonucleic acid*

**DNN** *deep neural network*

**FABI** *Forestry and Agricultural Biotechnology Institute*

**FFNN** *feed-forward neural network*

**GPT** *Generative Pre-trained Transformer*

**IoT** *Internet of Things*

**MLP** *multi-layer perceptron*

**MNIST** *Modified National Institute of Standards and Technology database*

**NLP** *natural language processing*

**PRC** *polymerase chain reaction*

**ReLU** *Rectified linear unit*

**RNN** *recurrent neural network*

**T5** *Text-to-Text Transfer Transformer*

**ViT** *Vision Transformer*

# u15048366_cos700_proposal.pdf

*by* AS (Andreas) Louw

**Submission date:** 10-Jul-2023 06:24PM (UTC+0200)
**Submission ID:** 2129188498
**File name:** u15048366_cos700_proposal.pdf (1.11M)
**Word count:** 4757
**Character count:** 25244

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA
Denkleiers • Leading Minds • Dikgopolo tša Dihlalefi

# COS700 Research Proposal

# Vision Transformers for Pest and Disease Detection

Andreas Louw
u15048366

u15048366@tuks.co.za

**Supervisor(s)**:
Dr. Anna Bosman
Mr. Arné Schreuder

**July 2023**

# Vision Transformers for Pest and Disease Detection


## Abstract

The ability for researchers and industry to identify and act upon observation of pests and diseases in the forestry and agriculture sector is important for the bio-security of a country. A reliable mechanism by which pest and diseases can be identified is crucial to ensure biosecurity. The image-processing research field has improved significantly in the last decade. Traditional approaches make use of *deep neural networks* (DNNs), in particular *convolutional neural networks* (CNNs). Much research has been done to extend on this field [1]. This includes pre-training, transfer learning, few-shot approaches, transformers and diffusion techniques. *Vision Transformers* (ViTs) pose a potential further improvement to the image classification problem for pests and diseases, especially where the visual similarities between species are fine-grained [2]. This research investigates different image classification techniques and architectures, comparing traditional CNNs to ViTs on a number of pest and disease datasets. All of the approaches make use of transfer learning from pre-trained, backbone networks. An empirical analysis is done to determine the effects of network size on the performance of these networks.



## Keywords:

artificial neural networks, deep neural networks, convolutional neural networks, transformers, vision transformers

# 1 Introduction

Domestic agriculture is important for any country to ensure food and material security for the populace. Pests and diseases are a threat to the security of food and material production. Work in the plant science field usually requires physical samples to be taken in the field. As pests and diseases can vary in types it is important to state that a full diagnoses cannot be made solely on images. Classifying pests and diseases with the help of image recognition is only a tool to help diagnoses. Depending on the situation further scientific methods should be followed to fully and reliably diagnose the pest and/or disease. This research is interested in comparing *convolutional neural network*s (CNNs) [1] with *Vision Transformer*s (ViTs) [2] when the goal is to detect pests and diseases on plants.

*Artificial neural network*s (ANNs) have been used in various ways to solve problems and speed up solving complex problems [3]. One of the main subfields of ANNs is deep learning [3]. One of the applications of DNNs is image recognition. Image recognition in artificial intelligence is the identifying and categorising of objects and/or patterns in an image or video. The current state of the art for image recognition tasks with deep learning is CNNs [1]. The neural network uses a convolutional layer that recognises patterns and features in the given image or video. A convolution is a mathematical operation that is preformed on two functions producing a third function. The third function expresses how the shape of the one function affects the shape of the other function.

An alternative to image recognition with CNN are ViT, ViTs are a deep learning model architecture that uses the transformer architecture for image recognition tasks [2]. The transformers neural network architecture was first introduced in the paper [4]. The transformers architecture was made popular by the *bidirectional encoder representations from transformers* (BERT) model used for natural language processing [5] whereafter it was adapted to image recognition tasks introduced in the paper [2].

The rest of the paper is set out as follows, the problem statement follows in Section 2 that elaborates on the motivation of this research. Section 3 contains the literature study conducted on how this problem is currently solved and investigates past and current research of ViTs. Section 4 sets out the methodology followed for this research. Section 5 outlines the planning for this research.

# 2 Problem Statement

The aim of this research is to discover whether ViT is a better approach than current CNN to train AI models with supervised learning on a dataset for pest and disease detection. The research aims at investigating the relevance of using ViT for early pest and disease detection in plant sciences. The main questions to be answered by this research are:

- How do ViTs compare to CNNs when dealing with pest and disease detection with images? Is the accuracy better or worse when both CNN and ViT is trained with the same dataset?
- Is ViT model more computational efficient than the CNN model?
  - Does ViT converge faster than CNN when training?
  - Does ViT use less memory than CNN when training?
- Investigate the effects of image resolution and scaling when training ViT.

# 3 Literature Study

The following section provides background for the research. The subsections that follows are: Section 3.1 that gives background on ANNs the building block of *artificial intelligence* (AI), with section 3.2 that gives background on *feed-forward neural network* (FFNN) and section 3.3 on training of neural networks. Section 3.4 introduces backpropagation. Section 3.5 covers CNNs that is the current state of the art for image processing. Section 3.6 discusses transformers that is the current state of the art for language models. Section 3.7 coversViTs that is a new architecture used for image processing, and Section 3.8 covers related work.

## 3.1 Artificial Neural Networks

This subsection introduces ANNs. ANNs were inspired by the function and structure of biological neural networks like the human brain [3]. This biological neural network consists of interconnected neurons.

The artificial neuron, illustrated in Figure 1, is the building block of ANNs. The artificial neuron is a mathematical model that simulates a biological neuron. The artificial neuron receives one or multiple input values,

gets the weighted sum of the inputs and passes the result through an activation function $f(x)$ along with the bias $b$ for the neuron that produces an output as shown in equation (1).

$$y = f(\Sigma xW + b) \tag{1}$$

Figure 1 illustrates how a artificial neuron works. Inputs $X_n$ along with a weight $W_n$ are summised. The summised total along with the bias is then used in an activation function, the activation function is the mechanism that decides if the neuron is activated or not. The output of the activation function is the output for the neuron.

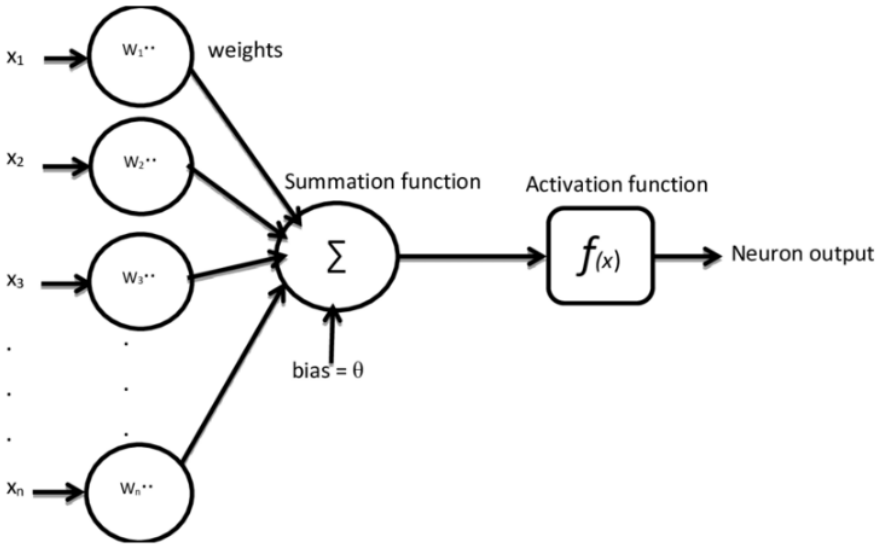


Figure 1: Artificial neuron illustration as taken from [6]. The figure illustrates how the summation (centre) of the inputs (left hand side) with weight then passed to the activation function (right hand side) to create the output.

The bias is a constant value added to the input of each neuron [7].

Activation functions are mathematical functions that add non-linearity to the output of neurons. The activation function determines if a neuron should be activated or not, based on the weighted sum of its inputs. Activation functions enable modelling of complex relationships in data The vanishing gradient problem is the phenomenon where the gradients of the loss functions become small. As a result convergence is slow and training performance is poor [8].

One of the most commonly used activation functions is the *Rectified linear unit* (ReLU) [9]. The ReLU formula is as shown in Eq (2).

$$f(x) = max\{0, x\} \tag{2}$$

The last major component of an ANN is the error function. An error function can also be referred to as loss function, cost function or objective function. An error function is a mathematical function that quantifies the error between the predicted output and the desired output. The error function is used to measure the performance of a model.

## 3.2 Feedforward Neural Network

This section introduces the necessary background for FFNN. FFNN is a neural network that is made of multiple layers with interconnected nodes. Information flows in a unidirectional manner form input to output layer. Each neuron calculates the weighted sum of inputs and applies the activation function this produces the output [10] as shown in Figure 2.

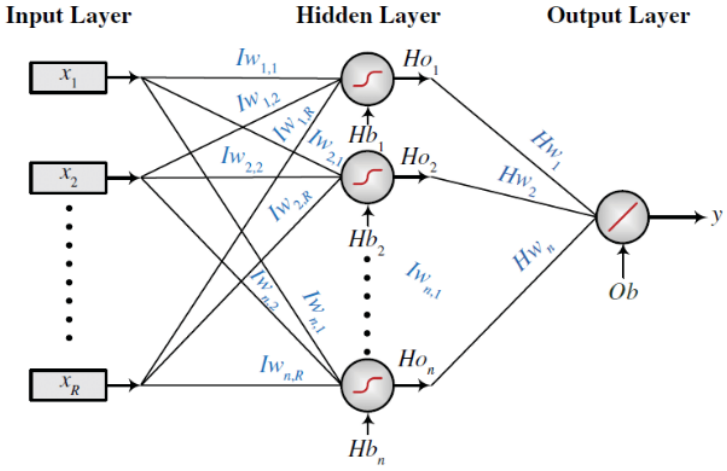


Figure 2: FFNN Model Architecture as taken from [11]. The input layers (left hand side), feed into the hidden layers (centre) with weights, in the hidden layer the activation function is applied to the input along with the weights, this produces the output in the output layer (right hand side).

### 3.3 Training

This subsection introduces the necessary background for Training of a neural network [12]. Training a neural network is the process of optimising the weights and biases of the model to learn the mapping between the inputs and outputs. During supervised training the weights and biases are adjusted iteratively based on the difference between the output and the desired outputs or actual values. The process of adjusting the weights and biases to minimise the difference between the output and desired output is known as backpropagation or optimisation. The goal of training a neural network is to minimise the error to allow the model to adjust well with new unseen data. There are several types of training; supervised learning, unsupervised learning, semi-supervised learning, reinforcement learning and transfer learning. Supervised learning uses datasets where the data is labelled, input-output pairs are used to train the model. The model learns to map the inputs to the outputs based on the given labels [13].

### 3.4 Backpropagation

This subsection introduces the necessary background on backpropagation. Backpropagation [14] is typically used when training CNN, it is a gradient-based optimisation algorithm that updates the weights of the network during training. Backpropagation calculates the gradient of the loss function with respect to the weights of the network and uses the gradient to update the weights so that the loss function is minimised.

### 3.5 Convolutional Neural Networks

This section introduces necessary background on CNNs. A CNNs are a type of deep learning model that is used for processing videos and images. CNNs are designed to extract and learn features from inputs such as images or videos. Currently CNNs is state of the art performance in image classification, object detection and image generation. CNNs consist out of the following concepts: convolutional layers, pooling layers, fully connected layers and backpropagation as illustrated in Figure 3. The following paragraph expands on these concepts.

Convolutional layers [15] applies convolutional operations on a given input. The convolutional operations that are preformed on the inputs include: computing dot products between filter and local input patch at each position, sliding small filters over the input image. These operations allow the

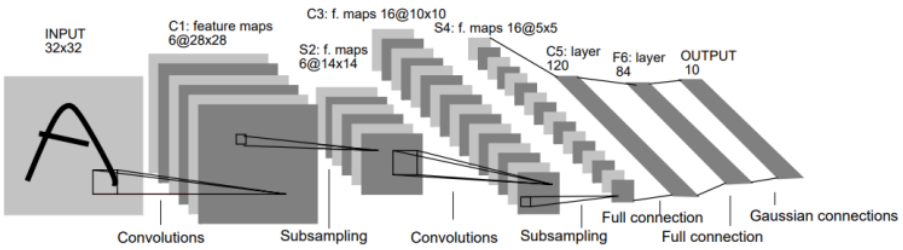


Figure 3: CNN model architecture as taken from [15]. An image is taken as input (left hand side), where the input is used in the convolutional layer (labelled as C for example C1: feature map) thereafter the out put of the convolutional layer is used in the sub-sampling layer (labelled as S for example S2 f. maps), the convolutional layer along with the sub-sampling layer is seen as one layer. The output of these layers are then used to make predictions.

network to learn features like edges, textures and patterns form the input. Pooling layers [16] are usually included in CNNs where the pooling layers downsample the feature maps created by the convolutional layers. Pooling aids in reducing the spatial dimensions of the feature maps created, this reduce the computational cost of the network and allows capturing of more abstract features. Fully connected layers [1] are traditional ANN layers that uses extracted features from the convolutional and pooling layers to make a final prediction. These predictions can be an image classification or an object detection.

### 3.6 Transformers

This subsection introduces the necessary background for transformers [4]. Transformers are a type of artificial deep learning model that is used in *natural language processing* (NLP) tasks. Transformers architecture relies on a self-attention mechanism that allows processing of input sequence in parallel in contrast to *recurrent neural network* (RNN) that process input sequentially. The self-attention mechanism is based of the concept of scaled dot-product attention [4]. The self-attention mechanism allows different weights to be assigned to different words in an input sequence, based on their relevance in the current context. The weights are used to calculate the weighted sum of the input words, this forms the basis of the models representations

and subsequent predictions. The use of self-attention allows transformers to capture long-range dependencies in sequence [4]. By adding to different parts of the input sequence with different levels of attention, it is possible to capture both local and global contextual information allowing modelling of complex patterns. Transformers use positional encoding techniques to encode the order of words since the word order does not have to be inherited like with RNNs. Positional encoding is typically achieved by adding fixed functions to the input embedding, to maintain the sequential order of input words during processing.

Most noticeable models using transformers are: BERT [5], *Generative Pre-trained Transformer* (GPT) [17] and *Text-to-Text Transfer Transformer* (T5) [18].

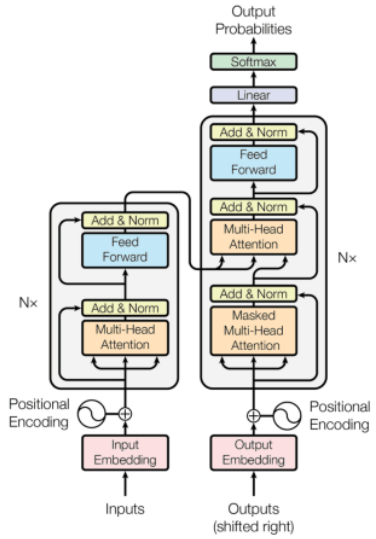


Figure 4: Transformer model architecture as taken from [4]. Text is given as input, the input is encoded, the encoded input is then assigned positional information that is then used as the input for encoder layer. The encoder (left hand side) layer maps all inputs to a abstract continuous representation that hold the learn information for the sequence. The decoder (right hand side) generates text as an output sequences by taking in the output from the encoding layer as well as the output generated by the decoder layer.

Figure 4 illustrates the Transformer model architecture, with the encoder on the left and decoder on the right. The received inputs are encoded to be

represented as a vector to capture the meaning. There after the embedded input goes through positional encoding, this allows the model to be aware of positional importance, that is needed in tasks like languages. Nx in the illustration are modules that can be stacked multiple times. The encoder consists out of mainly two layers: self-attention layer and the FFNN layer. The self-attention layer calculates the importance of a word by comparing it to other words in the input, attention scores are assigned based on importance. The output form the self-attention layer is fed into a FFNN that applies non-linear transformation to each position, allowing the model to learn complex relationships. The decoder also has both a self-attention layer and FFNN in addition to an encoder-decoder attention layer, this layer allows the model to creates output by taking the context into consideration. While training the model masking is used to stop the model form peeking at upcoming sequence, only allowing the model to have access to the previously generated output. The decoder crates output, one word at a time. At each generation the probability of the next word is predicted and the word with the highest probability is used as output, the predicted word is then used as input for predicting the next word.

### 3.7 Vision Transformers

This subsection introduces the necessary background for ViTs [2]. ViTs uses the Transformer architecture. ViT takes a new approach to image recognition tasks by breaking the image into small patches and treating each patch as a token. The patches are fed into a transformer encoder that processes it and gives a final output. ViT have been shown to have greater flexibility and scalability compared to CNNs.

ViTs consists out of the following concepts; Image Patching, Positional Embeddings, Transformer Encoder, Classification Head as illustrated in Figure 5.

The input image is divided onto patches that do not overlap, with a set fixed size, usually 16x16 or 32x32 pixels. Each patch is treated as a separate token and embedded into a one dimensional sequence.

Positional embedding encode the spatial information of the image patches and capturing their relative positions.

The patch tokens and positional embedding are processed by a series of transformer encoder blocks, that preform self-attention operations to capture local and global contextual information over the image.

The output of the transformer encoder is passed through a *multi-layer perceptron* (MLP) for final classification that predicts the labels for the image.

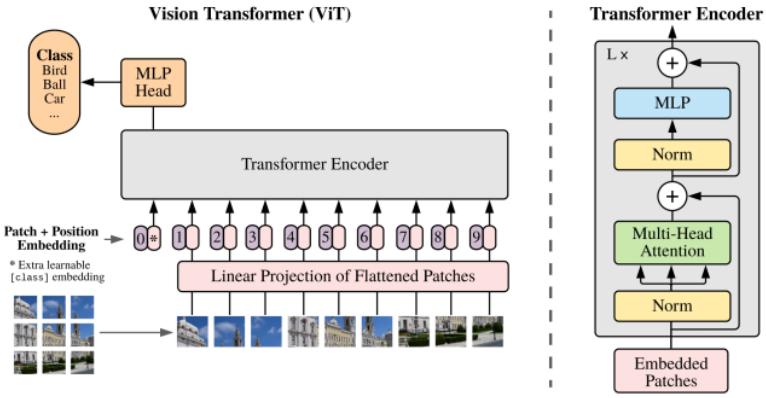


Figure 5: Vision transformer model architecture as taken from [2]. An image is given as input, the input is divided into patches (left hand side) and then flattened. positional encoding is then done on the linear projection of the flattened patches, this is then fed into the encoder (right hand side) that will produce an output.

## 3.8 Related Work

This section introduces related work in the field of pest and disease detection, highlighting the methods currently being used and some research that is done with pest and disease detection with AI. Currently there are several methods and technologies being used for pest and disease detection as provided by Martinelli et al.[19] and Li et al.[20] that provide a review of current methods and technologies.

Visual inspection is the process of trained personnel inspecting the plant for known pests and diseases and their symptoms [21]. Molecular Techniques involves techniques like *polymerase chain reaction* (PRC) and *Deoxyribonucleic acid* (DNA) sequencing [22]. These techniques involve analysing plant samples for presence of specific DNA sequences. Remote Sensing involves using satellite and drone imagery to detect changes in plant health, like changes in reflectance or thermal patterns [23]. Automated Monitoring Systems involves sensor networks and *Internet of Things* (IoT) devices to continuously monitor environment and plant health indicators [24]. Indicators that can be monitored are weather data like temperature, humidity, wind, ect.

Artificial intelligence, specifically deep learning is being used more commonly. Research done by Sladojvic et al. [25] explored plant disease detection with CNNs that is able to classify healthy and sick leaves, where it is able to

classify between 13 different diseases.

# 4 Methodology

This section introduces the methodology that the research will follow. Section 4.1 expands on the empirical process, Section 4.2 gives an overview of the datasets to be used, Section 4.3 gives a overview on the model configuration, Section 4.4 expand on the experiments to be conducted and Section 4.5 expands on how the statistical analysis will be handled for the research.

## 4.1 Empirical Process

An empirical process will be followed for the research as multiple experiments will be conducted to reject or verify the hypothesis about ViT and CNN. The experiments are as follows.

- Train a base CNN and ViT on the *Modified National Institute of Standards and Technology database* (MNIST), compare the outcomes of these models.
- Train a CNN and ViT on the pest and diseases datasets, compare the outcomes of these models.
- Compare the training time and performance of the models.
- Parameter tuning on the models to compare the effects.

## 4.2 Dataset

This section provides detail on the datasets used. Two datasets are used namely a *Forestry and Agricultural Biotechnology Institute* (FABI) dataset and the IP102 dataset. The IP102 dataset consists out of 75222 images that are labelled. An example image is given in Figure 6. These labels consist of both pests and diseases on 8 main classes of crop namely: Rice, Corn, Wheat, Beet, Alfalfa, Vitis, Citrus and Mango. The FABI dataset consists out of 2354 images, the image given in 7 gives an example. This dataset is specific to some of the pests and disease that are commonly found by FABI researchers.

Figure 6: Image taken from the IP102 dataset, presents a neoplocaederus

Figure 7: Image taken from the FABI dataset, presents shell lerp psyllid.

| **Dataset** | **Types** | **Attributes** | **Classes** | Resolution |
|---|---|---|---|---|
| MNIST | images | 70000 | 10 | 28x28 |
| IP102 | images | 75222 | 102 | 800x600 |
| FABI | images | 2354 | 8 | 1024x768 |

Table 1: Datasets

## 4.3 Models

The following subsection elaborates on the models for this research. There will be two main models that will be compared namely CNN and ViT. The configurations for the models are given in Table 2 and Table 3

The initial CNN configuration that will be used is given in Table 2 as defined in the original paper [26]:

| Parameter | Value |
|---|---|
| number of channels | 3 |
| embedding size | 64 |
| hidden sizes | [256, 512, 1024, 2048] |
| depths | [3, 4, 6, 3] |
| layer type | bottleneck |
| hidden activation function | 'relu' |
| downsample in fist stage | false |
| output features range | none |
| output indices | none |

Table 2: Table of Initial CNN Parameters.

The initial ViT configuration that will be used is given in Table 3 as defined in the original paper [2].

Both the IP102 and FABI datasets will be used for training and testing, the datasets will be split into 80% training and 20% testing.

## 4.4 Experiments

This subsection will give information on the experiments that will be conducted for this research. For each of these two models two experiments will be conducted, firstly a pre-trained model of each will be used and compared will both the IP102 and FABI datasets. The second will be training the models from scratch, where the IP102 dataset will be used to build a general model and then the FABI dataset used to specialise the model for FABI pest and disease detection. Experiments will consists of 10 runs for each model

| Parameter | Value |
|---|---|
| hidden size | 768 |
| number of hidden layers | 12 |
| number of attention heads | 12 |
| intermediate size | 3072 |
| hidden activation function | 'relu' |
| hidden dropout probability | 0.0 |
| attention dropout probability | 0.0 |
| initialiser range | 0.02 |
| layer normalisation epsilon | 1e-12 |
| image size | 224 |
| patch size | 16 |
| number channels | 3 |
| query key values bias | True |
| encoder stride | 16 |

Table 3: Table of Initial ViT Parameters.

and with each run the random seed will be changed. The run will run until completion with a maximum of 10 epochs as the stopping condition.

The expected output of each model on each run will be the average loss function for the run over all the iterations for that run, along with the run time, iteration number and step number.

The models performance will be determined as follows. The loss function used in the research is the cross entropy error. The classification accuracy will be measured at the end of each run on the testing data.

## 4.5 Statistical Analysis

A statistical analysis will be followed for analysing the experiments. Both trained models for CNN and ViT will given unseen testing data.

The outputs of the experiments will be taken then a comparison will

be made between the CNN and ViT models, this will be plotted as a graph. Possible statistical tests that will be used are *Analysis of Variance* (ANOVA) [27] statistical test, Mann-Whitney U test [28] and T-test [29], to determine statistical significance.

Each experiment will have 10 runs, a run will consists of unseen test data given as input to the trained model. Each run will have a maximum epoch stopping condition of 10.

Each run will produce a measurement of average accuracy for the run, the average loss over all the runs will give the average loss for the experiment.

# 5 Planning

This section highlights the research plan. Figure 8 outlines the timeline for the tasks. The main task to be done are: General research, writing of proposal, data collection and understanding, experiments and writing of final report.

| Task | Genral research | | | Write Proposal | Data | | | | Experiments | | | | | | Final Report Write up |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Find 4 papers on topic | Basic research on technology | | | Data Gathering | Data Understanding | Data Preperation | | Base CNN model | Base ViT model | Extend CNN with pest and disease | Extend ViT with pest and disease | Parameter tuning | |
| Weeks | 4 | 1 | 3 | 4 | 3 | 1 | 1 | 1 | 13 | 1 | 2 | 2 | 3 | 5 | 7 |
| 03/04/2023 | | | | | | | | | | | | | | | |
| 10/04/2023 | | | | | | | | | | | | | | | |
| 17/04/2023 | | | | | | | | | | | | | | | |
| 24/04/2023 | | | | | | | | | | | | | | | |
| 01/05/2023 | | | | | | | | | | | | | | | |
| 08/05/2023 | | | | | | | | | | | | | | | |
| 15/05/2023 | | | | | | | | | | | | | | | |
| 22/05/2023 | | | | | | | | | | | | | | | |
| 29/05/2023 | | | | | | | | | | | | | | | |
| 05/06/2023 | | | | | | | | | | | | | | | |
| 12/06/2023 | | | | | | | | | | | | | | | |
| 19/05/2023 | | | | | | | | | | | | | | | |
| 26/06/2023 | | | | | | | | | | | | | | | |
| 03/07/2023 | | | | | | | | | | | | | | | |
| 10/07/2023 | | | | | | | | | | | | | | | |
| 17/07/2023 | | | | | | | | | | | | | | | |
| 24/07/2023 | | | | | | | | | | | | | | | |
| 31/07/2023 | | | | | | | | | | | | | | | |
| 07/08/2023 | | | | | | | | | | | | | | | |
| 14/08/2023 | | | | | | | | | | | | | | | |
| 21/08/2023 | | | | | | | | | | | | | | | |
| 28/08/2023 | | | | | | | | | | | | | | | |
| 04/09/2023 | | | | | | | | | | | | | | | |
| 11/09/2023 | | | | | | | | | | | | | | | |
| 18/09/2023 | | | | | | | | | | | | | | | |
| 25/09/2023 | | | | | | | | | | | | | | | |
| 02/10/2023 | | | | | | | | | | | | | | | |
| 09/10/2023 | | | | | | | | | | | | | | | |
| 16/10/2023 | | | | | | | | | | | | | | | |
| 23/10/2023 | | | | | | | | | | | | | | | |
| 30/10/2023 | | | | | | | | | | | | | | | |

Figure 8: Planning Table

## References


[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[2] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[3] A. K. Jain, J. Mao, and K. M. Mohiuddin, "Artificial neural networks: A tutorial," *Computer*, vol. 29, no. 3, pp. 31–44, 1996.

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[6] J. A. Yacim and D. G. B. Boshoff, "Impact of artificial neural networks training algorithms on accurate prediction of property values," *Journal of Real Estate Research*, vol. 40, no. 3, pp. 375–418, 2018.

[7] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction*, vol. 2. Springer, 2009.

[8] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 6, no. 02, pp. 107–116, 1998.

[9] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814, 2010.

[10] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

[11] S. Razavi and B. A. Tolson, "A new formulation for feedforward neural networks," *IEEE Transactions on neural networks*, vol. 22, no. 10, pp. 1588–1598, 2011.

[12] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[13] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image segmentation using deep learning: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 7, pp. 3523–3542, 2021.

[14] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[15] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[16] P. Sermanet and Y. LeCun, "Traffic sign recognition with multi-scale convolutional networks," in *The 2011 international joint conference on neural networks*, pp. 2809–2813, IEEE, 2011.

[17] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[18] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 5485–5551, 2020.

[19] F. Martinelli, R. Scalenghe, S. Davino, S. Panno, G. Scuderi, P. Ruisi, P. Villa, D. Stroppiana, M. Boschetti, L. R. Goulart, *et al.*, "Advanced methods of plant disease detection. a review," *Agronomy for Sustainable Development*, vol. 35, pp. 1–25, 2015.

[20] L. Li, S. Zhang, and B. Wang, "Plant disease detection and classification by deep learning—a review," *IEEE Access*, vol. 9, pp. 56683–56698, 2021.

[21] R. Balodi, S. Bisht, A. Ghatak, K. Rao, *et al.*, "Plant disease diagnosis: technological advancements and challenges," *Indian Phytopathology*, vol. 70, no. 3, pp. 275–281, 2017.

[22] M. M. López, P. Llop, A. Olmos, E. Marco-Noales, M. Cambra, and E. Bertolini, "Are molecular tools solving the challenges posed by detection of plant pathogenic bacteria and viruses?," *Current issues in molecular biology*, vol. 11, no. 1, pp. 13–46, 2009.

[23] F. Baret, V. Houlès, and M. Guerif, "Quantification of plant stress using remote sensing observations and crop models: the case of nitrogen management," *Journal of experimental botany*, vol. 58, no. 4, pp. 869–880, 2007.

[24] I. Buja, E. Sabella, A. G. Monteduro, M. S. Chiriacò, L. De Bellis, A. Luvisi, and G. Maruccio, "Advances in plant disease detection and monitoring: From traditional assays to in-field diagnostics," *Sensors*, vol. 21, no. 6, p. 2129, 2021.

[25] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, "Deep neural networks based recognition of plant diseases by leaf image classification," *Computational intelligence and neuroscience*, vol. 2016, 2016.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[27] T. K. Kim, "Understanding one-way anova using conceptual figures," *Korean journal of anesthesiology*, vol. 70, no. 1, pp. 22–26, 2017.

[28] H. B. Mann and D. R. Whitney, "On a test of whether one of two random variables is stochastically larger than the other," *The annals of mathematical statistics*, pp. 50–60, 1947.

[29] Student, "The probable error of a mean," *Biometrika*, vol. 6, no. 1, pp. 1–25, 1908.

# A Appendix

## Acronyms

**AI** *artificial intelligence*

**ANN** *artificial neural network*

**ANOVA** *Analysis of Variance*

**BERT** *bidirectional encoder representations from transformers*

**CNN** *convolutional neural network*

**DNA** *Deoxyribonucleic acid*

**DNN** *deep neural network*

**FABI** *Forestry and Agricultural Biotechnology Institute*

**FFNN** *feed-forward neural network*

**GPT** *Generative Pre-trained Transformer*

**IoT** *Internet of Things*

**MLP** *multi-layer perceptron*

**MNIST** *Modified National Institute of Standards and Technology database*

**NLP** *natural language processing*

**PRC** *polymerase chain reaction*

**ReLU** *Rectified linear unit*

**RNN** *recurrent neural network*

**T5** *Text-to-Text Transfer Transformer*

**ViT** *Vision Transformer*

# u15048366_cos700_proposal.pdf

ORIGINALITY REPORT

| 17% | 13% | 7% | 4% |
|---|---|---|---|
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

| | | |
|---|---|---|
| 1 | huggingface.co<br>Internet Source | 2% |
| 2 | trepo.tuni.fi<br>Internet Source | 1% |
| 3 | baptiste-wicht.com<br>Internet Source | 1% |
| 4 | www.diva-portal.org<br>Internet Source | 1% |
| 5 | Partha Pratim Ray. "ChatGPT: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope", Internet of Things and Cyber-Physical Systems, 2023<br>Publication | 1% |
| 6 | Submitted to University of Wales Swansea<br>Student Paper | <1% |
| 7 | Submitted to SSN COLLEGE OF ENGINEERING, Kalavakkam<br>Student Paper | <1% |

| | | |
|---|---|---|
| 8 | itmmconf.tsu.ru<br>Internet Source | <1 % |
| 9 | repositorio.unicamp.br<br>Internet Source | <1 % |
| 10 | soka.repo.nii.ac.jp<br>Internet Source | <1 % |
| 11 | Submitted to Ohio University<br>Student Paper | <1 % |
| 12 | Submitted to SUNY Canton<br>Student Paper | <1 % |
| 13 | Submitted to National School of Business Management NSBM, Sri Lanka<br>Student Paper | <1 % |
| 14 | Submitted to University of Leeds<br>Student Paper | <1 % |
| 15 | XingLong Wang, Wei Lu. "Automatic Text Summarization Technology of Keyword Replacement Based on Seq2Seq", 2021 7th International Conference on Systems and Informatics (ICSAI), 2021<br>Publication | <1 % |
| 16 | Ankita Dey, Sreeraman Rajan, George Xiao, Jianping Lu. "Fall Event Detection using Vision Transformer", 2022 IEEE Sensors, 2022<br>Publication | <1 % |

| | | |
|---|---|---|
| 17 | Submitted to University of Birmingham<br>Student Paper | <1 % |
| 18 | Submitted to University of East London<br>Student Paper | <1 % |
| 19 | d-nb.info<br>Internet Source | <1 % |
| 20 | discovery.researcher.life<br>Internet Source | <1 % |
| 21 | Thi Nhan Nguyen, Van Cong Le, Se-Hyeon Noh, Chan-Woo Park. "ANN-based Prediction of Nusselt Number and Stored Energy in PCM Heat Exchanger for Solar Heat Storage", 2022 IEEE/ACIS 7th International Conference on Big Data, Cloud Computing, and Data Science (BCD), 2022<br>Publication | <1 % |
| 22 | M.E. Buckley, S.B. Wicker. "The design and performance of a neural network for predicting turbo decoding error with application to hybrid ARQ protocols", IEEE Transactions on Communications, 2000<br>Publication | <1 % |
| 23 | Thanh-Dat Truong, Ravi Teja Nvs Chappa, Xuan-Bac Nguyen, Ngan Le, Ashley P.G. Dowling, Khoa Luu. "OTAdapt: Optimal Transport-based Approach For Unsupervised | <1 % |

Domain Adaptation", 2022 26th International Conference on Pattern Recognition (ICPR), 2022
Publication

| | | |
|---|---|---|
| 24 | assets.researchsquare.com<br>Internet Source | <1 % |
| 25 | www.arxiv-vanity.com<br>Internet Source | <1 % |
| 26 | Anthony Guillard, Paul Thevenon, Carl Milner. "Using convolutional neural networks to detect GNSS multipath", Frontiers in Robotics and AI, 2023<br>Publication | <1 % |
| 27 | Chen Lu, Zhenya Wang, Bo Zhou. "Intelligent fault diagnosis of rolling bearing using hierarchical convolutional network based health state classification", Advanced Engineering Informatics, 2017<br>Publication | <1 % |
| 28 | Jiang Lu, Jie Hu, Guannan Zhao, Fenghua Mei, Changshui Zhang. "An in-field automatic wheat disease diagnosis system", Computers and Electronics in Agriculture, 2017<br>Publication | <1 % |
| 29 | dokumen.pub<br>Internet Source | <1 % |

downloads.hindawi.com

| | | |
|---|---|---|
| 30 | Internet Source | <1% |
| 31 | kth.diva-portal.org<br>Internet Source | <1% |
| 32 | www.hindawi.com<br>Internet Source | <1% |
| 33 | www.researchgate.net<br>Internet Source | <1% |
| 34 | Samantha MacDougall, Fatih Bayansal, Ali Ahmadi. "Emerging Methods of Monitoring Volatile Organic Compounds for Detection of Plant Pests and Disease", Biosensors, 2022<br>Publication | <1% |
| 35 | Yi-Heng Zhu, Dong-Jun Yu. "ULDNA: Integrating Unsupervised Multi-Source Language Models with LSTM-Attention Network for Protein-DNA Binding Site Prediction", Cold Spring Harbor Laboratory, 2023<br>Publication | <1% |
| 36 | arxiv.org<br>Internet Source | <1% |
| 37 | doctorpenguin.com<br>Internet Source | <1% |
| 38 | dspace.daffodilvarsity.edu.bd:8080<br>Internet Source | <1% |

| | | |
|---|---|---|
| 39 | dspace5.zcu.cz<br>Internet Source | <1 % |
| 40 | open.library.ubc.ca<br>Internet Source | <1 % |
| 41 | patentimages.storage.googleapis.com<br>Internet Source | <1 % |
| 42 | proceedings.neurips.cc<br>Internet Source | <1 % |
| 43 | researchmgt.monash.edu<br>Internet Source | <1 % |
| 44 | scholar.sun.ac.za<br>Internet Source | <1 % |
| 45 | uu.diva-portal.org<br>Internet Source | <1 % |
| 46 | vdoc.pub<br>Internet Source | <1 % |
| 47 | vtechworks.lib.vt.edu<br>Internet Source | <1 % |
| 48 | www.diva-portal.se<br>Internet Source | <1 % |
| 49 | Mehdi Mohammadi, Ala Al-Fuqaha, Sameh Sorour, Mohsen Guizani. "Deep Learning for IoT Big Data and Streaming Analytics: A | <1 % |

Survey", IEEE Communications Surveys & Tutorials, 2018
Publication

| | | |
|---|---|---|
| 50 | Minh-Tri Nguyen, Duong H. Le, Takuma Nakajima, Masato Yoshimi, Nam Thoai. "Attention-Based Neural Network: A Novel Approach for Predicting the Popularity of Online Content", 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), 2019<br>Publication | <1 % |
| 51 | scholar.uwindsor.ca<br>Internet Source | <1 % |

| | | | |
|---|---|---|---|
| Exclude quotes | On | Exclude matches | Off |
| Exclude bibliography | On | | |