Using a Smart Chatbot System as a Communication Tool for Campus Navigation

Goodman Dimo, Tshwane University of Technology, South Africa
Joey Jansen van Vuuren, Tshwane University of Technology, South Africa
Anna-Marie Jansen van Vuuren, Tshwane University of Technology, South Africa

The Asian Conference on Media, Communication & Film 2022
Official Conference Proceedings

Abstract
First-year students and visitors to South African universities often lose their way. This is due to widespread areas and complex routes or pathways on campuses. The experience can be intimidating and confusing to students and visitors who ask for directions from senior students or university personnel at helpdesks. This paper presents a potential answer to this dilemma – a hybrid chatbot system named SoshaMapBot. This chatbot was developed and tested on the Soshanguve campus of the Tshwane University of Technology in South Africa. The SoshaMapBot uses the entity recognition technique of natural language processing to process the written texts that the user types into the chatbot. The SoshaMapBot asks the users to upload photographs of their surroundings to the program via text. The chatbot then captures these images as inputs and returns textual responses with directions to the user. This study evaluated the proposed chatbot to confirm whether using image processing in chatbots is a viable and sustainable methodology. During the evaluation process of the developed chatbot, the researchers found that the developed chatbot exhibited a 75% accuracy rate in determining where the students or visitors find themselves on campus and giving the correct directions to them. The research found that image processing is a feasible way to develop chatbots and the findings are explained in this paper. However, image processing must be combined with the correct choice of technology, algorithms, and methodologies. Therefore, by developing this chatbot, this study contributes to communication technology for enhanced communication techniques.

Keywords: Chatbot, Campus, Navigation, Communication, Communication Techniques, Communication Technology, Image Processing, Location Finding, Algorithm

Introduction

A chatbot can be defined as a "computer program designed to simulate conversation with human users, especially over the internet" (Adamopoulou & Moussiades, 2020). The computer program combines artificial intelligence (AI) and natural language processing (NLP) to understand the user's questions and automatically respond to them (by using text-like human conversation). NLP is a technology that uses machine learning algorithms to give a machine the ability to understand, learn and make sense of 'human' text. The machine can then respond to the text or voice data with text or speech of their own that is understandable to humans (Adamopoulou & Moussiades, 2020). Chatbots are thus very convenient as the chatbots can be used to respond to the user's questions quickly (Adamopoulou & Moussiades, 2020). (Adamopoulou & Moussiades, 2020). Machine learning algorithms (also called reinforcement learning algorithms) allow the chatbot to learn new data and information over time. This results in the responses' accuracy also increasing over time (Singh et al., 2019).

This study elaborated on in this paper investigated the development of a chatbot. The chatbot aim is to support new students and visitors to find specific locations on a large university campus where there is no information or guidance available on Google maps or other similar programs. ShoshaMapbot, the chatbot developed in this study, will guide the user when navigating to the destination they are looking for on campus. The chatbot uses a photo to determine the user's position on the campus. A photo, also called an image, consists of a number of pixels. Every pixel has a specific shade, opacity or colour. The number of pixels in the image is dependent on the height and width of the image. An image of 500x400 will have 20000 pixels. These pixels are used for image processing. Image processing converts an image into a digital format (pixels) that can be used to enhance the image or get specific information from it (Arthur et al., 2017, p. 123). Our study's chatbot uses image processing to predict the user's current location, and identifies the required destination from the conversation on the chatbot.

Chatbots do not often use AI with image processing. This can be attributed to the challenges in implementing the combinations of natural language and image processing (Adamopoulou & Moussiades, 2020). This study focused on the development of a chatbot and how image processing can be incorporated into chatbot development. The image processing in the chatbot is used to predict the user's current location and provide campus navigation.

The next section of this paper provides the literature review we conducted before developing the chatbot. The methodology follows the literature review. Then we provide a discussion on the development and implementation of the chatbot (ShoshaMapBot). A presentation of the results follows, and then the conclusion is presented.

Literature Review

The following studies were found in our survey of the literature. These studies identify the algorithms and techniques incorporated in image-processing chatbots.

Dechert et al. (2019) developed a disembodied conversational agent (CA) chatbot. This chatbot was aimed at collecting geotagged images called Dengue Detector (DD). The Dengue Detector receives and prompts the user's location and provides instruction indicators for taking images. The Dengue Detector is developed using DialogFlow in conjunction with basic Natural Language Processing (NLP) techniques, intent recognition, and the Facebook application programming interface (API) for the user interface.

Lee et al. (2019) developed a web-based chatbot that provides a so-called 'social magazine' consisting of video content. Their chatbot feeds on news from various social media platforms and then creates a video or magazine based on the results of the 'social media news analysis'. The video is created by combining sets of images, repetition, and video effects such as fade-in and fade-out techniques. The developers then used the Fast Forward MPEG (FFMPEG) library to create background music synthesis for the videos.

Another interesting system quite similar is a video chatbot developed by Li et al. (2016). This chatbot system allows users to comment on each other's posts using video and has a text-based feature as an added functionality. Users can extract and search visual features from videos. In order to extract visual features of the video, they used convolutional neural networks (CNN), and for searching through the video, they used the approximate nearest neighbour (ANN). The chatbot could only be accessed on a computer or laptop with a web camera.

Pardasani et al. (2018) developed a hand-gesture recognition chatbot model that can be used to communicate with deaf and hard-of-hearing people. They aimed to fill the gap between the use of voice chatbots amongst abled and deaf, and hard-of-hearing people.

Sathit developed a financial trading system composed of an image processing service that checked whether the customer had uploaded a valid image for the transfer. All the responses in the database were predefined, and the system did not produce any new responses outside of the predefined scope (Prasomphan, 2019).

In the literature that we studied and summarised briefly above, we have observed that we could develop a unique chatbot different from the ones developed as part of the abovementioned studies. The main novelty of our research with the SoshaMapBot system is that this study introduces a unique technique for developing task-based chatbots that incorporates image processing. Additionally, the explicit difference between the systems developed in previous research and this study's chatbot is that none of the prior chatbots uses image processing to predict the user's locations. When an image processing technique is applied in the previous studies, it does not have feature matching in which at least two images are compared.

Methodology

This study adopted the design science methodology. Design science research focuses on the innovation and utility of the artefact produced, which in this study is the image-processing chatbot. The artefact contributes as an innovative solution towards implementing chatbots that can read images and use information retrieved from an image to determine the current location. Therefore, the developed artefact is evaluated based on meeting specific requirements and serving a human purpose (Hevner et al., 2004; Jansen van Vuuren et al., 2016). The outcomes of a design science-based research artefact can vary depending on the problem being addressed. As such, the contribution of an artefact is based on the usability of an existing artefact for a new purpose or in a new environment or context (Cannas & Gosling, 2021).

Figure 1 (adapted from Jansen van Vuuren, Grobler, Leenen, Chan, and Dawood, 2016) demonstrates the adopted methodology's iteration process in design science format. Stage 1 in Figure 1 highlights the conceptual and explanatory research, including requirements extraction and literature study. Stage 2 highlights a description of the model, an evaluation of the results, and the maintenance process (Jansen van Vuuren et al., 2016).

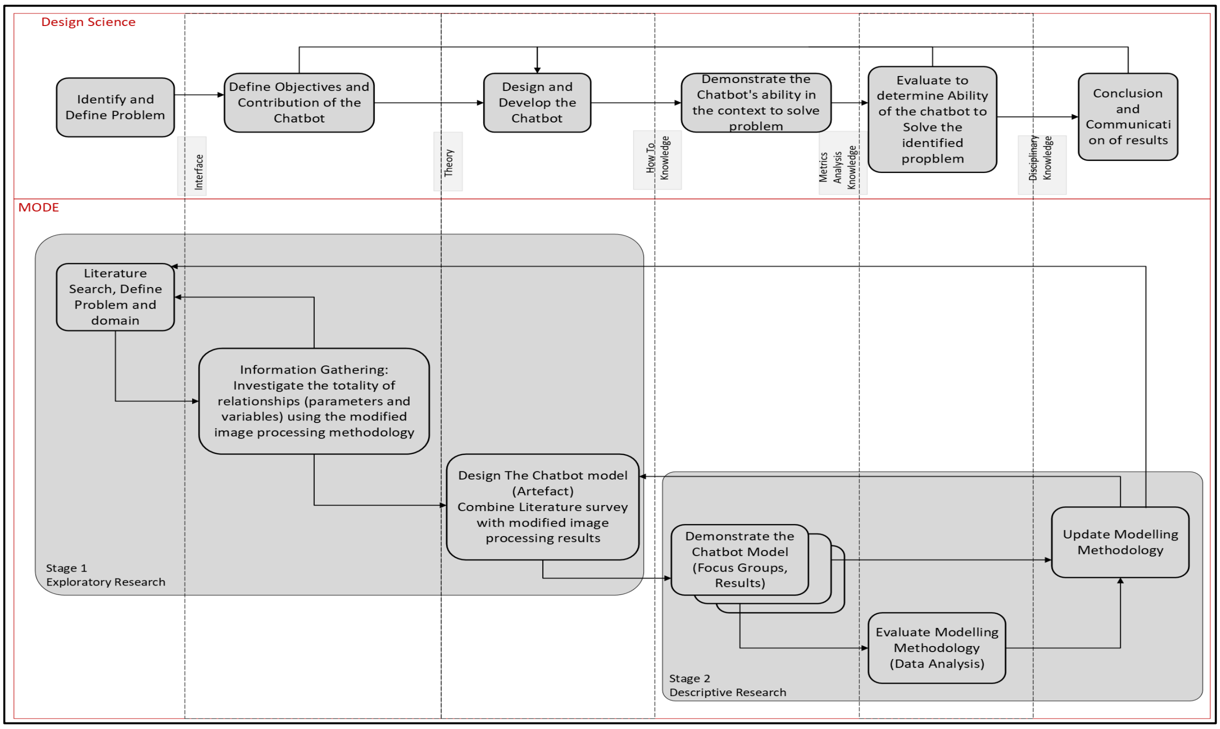

Figure 1: *Methodology (Jansen van Vuuren et al., 2016)*

This study also used machine learning methods in the image processing stage of the creation of the chatbot artefact. This method is proposed by Arthur et al. (2017) and includes *dataset creation, digitalisation*, *background subtraction (interpolation)*, *element segmentation and cropping,* and *feature matching*. The image processing technique used to find matching features of the images was adapted to use feature matching instead of feature extraction. Feature matching is "the act of recognizing features of the same object across images with slightly different viewpoints" (Roelke, 2013). This feature-mapping step uses the k-nearest neighbour (KNN) classification algorithm and Euclidean distance. According to Raschka (2018, p. 3), "nearest neighbour algorithms are among the 'simplest' supervised machine learning algorithms" that can be used in the field of pattern recognition. A supervised machine learning algorithm "relies on labelled input data to learn a function that produces appropriate output when given unlabelled data" (Harrison, 2018). The technique applies the Gaussian elimination rule to the image pixel to find and verify matching features within the images when they are being compared.

The next sections discuss how these methods are used to develop the image processing model that takes an image as input and predicts the user's current location.

Development of the SoshaMapBot

The ShoshaMapBot is an image processing-based chatbot with five components: mobile application, bot service, multimedia database, image processing (IP) service, and digital file storage. Authentication and authorisation information security protocols are necessary to ensure the chatbot communicates with the correct users and does not mix-match responses. A unique username allocation authorisation process was used with the user's internet protocol (IP) address as the username to ensure that when the location identification results come back from the IP service, they get sent to the correct user. This username is allocated at every session start and revoked at every session end. A new IP username is allocated to the user during every session. This ensures that the response is sent to the user with a specific IP address and immediately deletes the records after responding to avoid duplications and miss matching results. The architecture of the ShoshMapBot is shown in Figure 2.

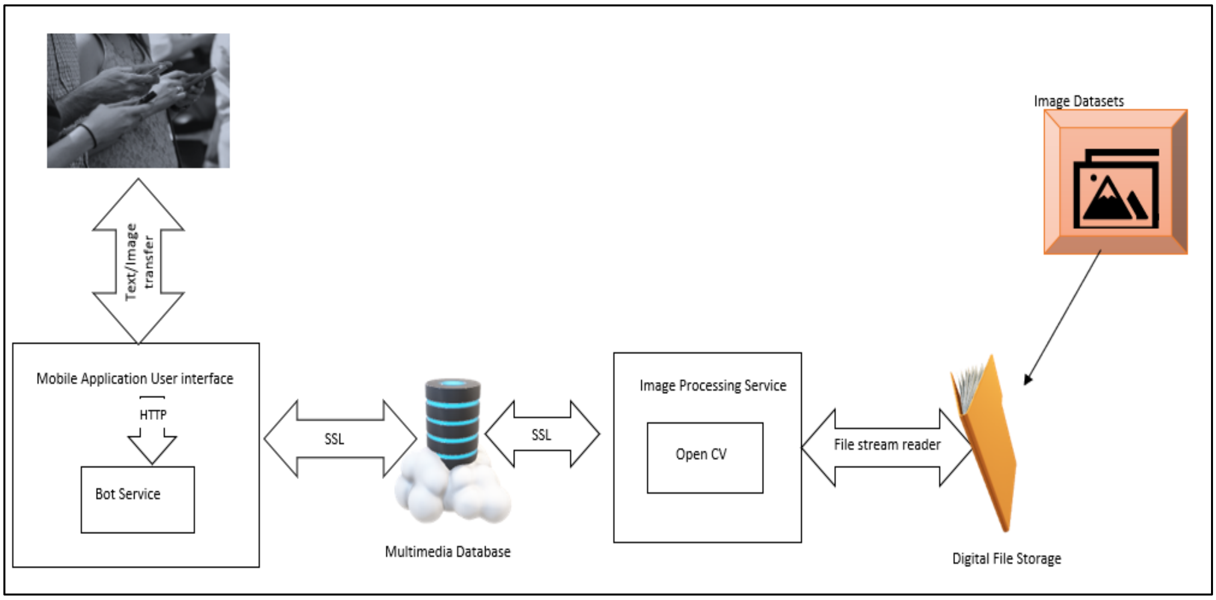

Figure 2: *Architecture of the ShoshaMapbot*

User Interface

The SoshaMapBot interface is developed for communication between the user and the chatbot service. This self-created mobile application interface is developed and integrated with the chatbot service, thus allowing the user to initiate and maintain communication through the mobile application service. The user interacts with the service using text and can upload the image using an attachment button. The user interface is shown in Figure 3.

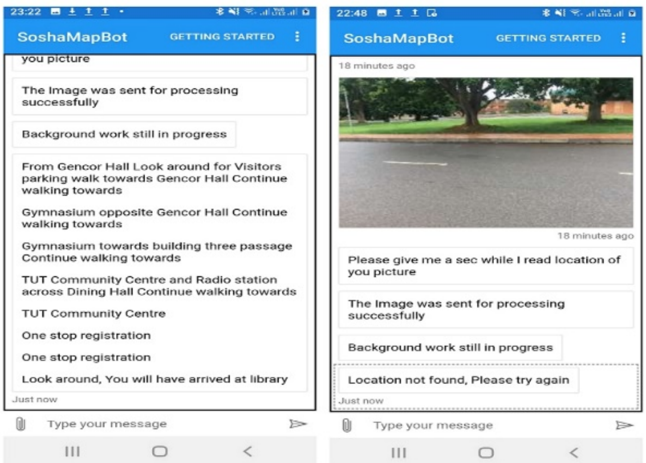

Figure 3: *SoshaMapBot – user interface with directions*

The user interacts with the chatbot to find the campus directions, instead of the chatbot asking for the user's current location. The chatbot requests an image from the user of their location and sends it to the image processing model in the background. When the image processing model has completed its tasks, results are returned to the user as directions or information messages notifying the user that the location could not be found.

Bot Service

A bot service is a web service that can be integrated into many applications to provide front-end functionality. In the case of ShoshaMapbot, the bot service is a web service with a single line of HTTP WebView referring to the bot service. Microsoft's Azure Bot Service was used to create the bot service. The SoshaMapBot chatbot system needs an image to determine the user's location. The bot service provides the conversational flow between the user and the chatbot. The user uses a mobile device to capture the image, which is then uploaded to the chatbot system. The service is deployed on the Microsoft Azure cloud platform as a web service accessed on the mobile application through a webview URL that provides a user-friendly interface to the user.

In order to make SoshaMapBot understand the user's requests for the location, the bot uses the named-entity recognition NLP technique. The named-entity recognition algorithm identifies the destination location from the user's response by performing a dictionary word search of specific locations of the campus stored in the location dictionary. It then computes directions from the source to a destination following the user's image upload.

Multimedia Database

The multimedia database resides at the heart of SoshaMapBot and is used as an intermediate point of communication between the image processing service and the bot service. It consists of four databases and five collections. The main functionality of the database is to store statically constructed information on campus areas. Unique symbols, identifiers, and names were used to identify the proposed areas, directions, and paths. The data collected must be dynamic to make provision for the performance of quick changes or improvements on the system if necessary. The multimedia database was built using MongoDB. The bot service, deployed on the Microsoft Azure cloud platform, performs requests using secure socket layer (SSL) protocol calls from the Microsoft Azure servers to the MongoDB servers to retrieve and store specific data. Each component sets up a communication pipeline for the communications.

Image Processing Service

Dataset Creation and Digitalisation

Dataset creation and digitalisation were used to get the necessary dataset for the ShoshaMapBot prototype that could be used for the development and training of the model. A wide range of datasets was required for learning the chatbot system. The images were collected, filtered, and sorted in a way deemed fit for the development process. The development process was done in five stages: data collection, image collection, dataset filtering, file storage, and database structure.

During data collection, information about the campus, especially those normally visited areas, was collected for use in the prototype. That included information such as campus buildings, monuments, campus area spaces, and campus residences. Over one hundred names of campus areas and spaces were recorded. These datasets and a dictionary were used to develop the prototype and for training of the model.

For the image collection stage, photographs (images) were taken at the various campus areas previously identified in the data collection stage. Originally, 591 images were captured using these cell phones: Samsung A2 core, Samsung A10, and Apple iPhone 6. The images collected included all the identified campus areas and were captured from various angles. The reason for using various devices and capturing various angles was to increase the accuracy of the image or critical pointer detection process using various image pixels and to make provision for various mobile devices that will use the system. The pixels of each image are calculated by multiplying the width and the height of the image. A pre-written Python script was used to perform the calculations.

For the dataset filtering, images were manually filtered by creating collages of images with various angles in the same area, reducing the number of images stored. Originally 591 images were captured. The collaging technique reduced one group of images from 566 to four images. In order to ensure that this technique was effective, it was tested with the training model to determine if the images could still be recognised. An example of the collaged images is provided in Figure 4. After the data filtration step, there were 65 records of campus areas and space records left.

The images were then stored in the virtual file storage. After filtering, the images collected, with the location areas, were stored in a non-relational database, MongoDB. The database also included the names of campus locations and the directions for various route options from one campus area to another.

Figure 4: *DataSet Sample – Gencor hall of TUT Soshanguve Campus*

Background Subtraction (Interpolation)

In order to predict the user's current location, interpolation handles the raw image uploaded by the user on request of the chatbot. The interpolation converts the raw image into a machine-usable format (binary format). This stage includes data import and background subtraction. The data import step occurs on the front end of the mobile application via the bot service.

The first background subtraction is performed on the image-processing service. The data import process is done when the chatbot asks the user to upload the image using the standard Microsoft bot service attachment prompt.

The second background subtraction stage starts with the conversion of the image to greyscale. The subtraction includes the application of functions to turn the image to a greyscale that will reduce its size. The width and height are reduced to 50% of the original size to ensure that the image processing service is able to ingest the image. An example of such an image is given in Figure 5.

Figure 5: *Black and white image during interpolation*

Element Segmentation and Cropping

For image processing, the images are divided into parts to create segments and regions based on interchangeable critical pointers of the independent images to enable the comparisons that will lead to image recognition. Image pixels are cropped to eliminate corresponding edges. OpenCV's scale-invariant feature transform (SIFT) method of detecting and computing image segments was used for this process. This process also includes an analysis of the image pixels. Figure 6 shows an image after element segmentation and cropping was done. The resulting image will be a (M x N) matrix with a set of integer values.

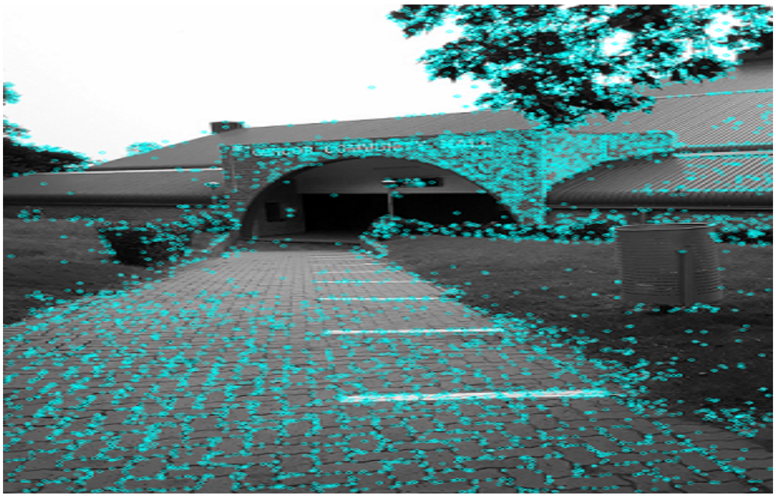

Figure 6: *Segmented image*

Feature Matching

Feature matching is the most crucial step in the methodology for this study. It ensures that images are matched correctly to predict the user's location. This step has been modified from the original methodology used by Arthur et al. (2017) and uses feature matching instead of feature extraction to satisfy this study's aim. The features required are extracted from the image and matched with the dataset images for training the datasets. This is performed by identifying and marking features matched in the compared images. The matching is computed using the KNN algorithm provided by the OpenCV library. The Euclidian distance formula is also utilised to increase the prediction probability, and a K scalar multiple reduces the distance second image.

A test was done to verify if the image processing procedure would give the correct results. Figure 7 indicates the test to see if feature mapping was done correctly.

Figure 7: *Colour images featuring the 'mapping test'*

When the user-coloured image is provided to the system, the image goes through the interpolation. Figure 8 presents the results of the image processing service after processing the images provided in Figure 7. In this test case, the system possibly would identify the grey images as the same. However, one aspect was different. The purple circle on the left of Figure 7 shows that the two pictures are not identical and is the prominent significant feature tested. The system had to identify that one of the features differs between the two pictures, and therefore it is not the same location. The test's goal was for the service to accurately forecast that the features differ; thus, the areas in the photos were not in the same place. As in this case, and in some other cases, the image processing service cannot find the difference, as seen in Figure 7.

The training model found as many matches as possible but not the critical feature in question. Even while the differences are visible, they are insufficient to provide a clear indication or forecast whether the images provided are from the same place. In such circumstances, using the Gaussian elimination to test and confirm the coordinate is extremely valuable. This validity testing should be enough to ensure that SoshaMapBot's image processing training model can detect and predict whether the image's location matches the user's current location.

Figure 8: *Image processing results*

Results

The developed artefact proved usable and innovative, as it injected a self-developed chatbot application to provide campus directions. Based on the outputs of the chatbots, the image processing model was then evaluated based on the F1 score to determine the model's usability and suitability in chatbot development. Following several tests of the chatbot, we found that the developed image processing model exhibits 50% true positives, 0% false positives, 33% false negatives, and 17% true negatives using the F score for validation.

The F1 score ranges from 0 to 1, in which 0 means the model is bad and 1 means the model is good. Therefore, the closer the results are to 1, the better the model. Therefore, observed from the above F1-score calculations resulting in 0.75 (75%) means that the performance of the image processing service prototyped in this study is fairly good, and it is able to send and retrieve correct information as per information perspective requirement from the chatbot knowledge base. This requirement entails that the information gathered by the chatbot must be appropriate and meets the information needs of the users (Peras, 2018).

However, the system is slightly slow, and another type of algorithm must be considered to make the feature selection due to the time used for single images to be analysed on a virtual server. An algorithm that can analyse images in a non-linear manner would be ideal.

This study used the scale-invariant feature transform (SIFT) algorithm provided by the OpenCV library in addition to the KNN algorithm. SIFT is an image processing algorithm to extract image features. We learnt that the SIFT algorithm is inefficient for SoshaMapBot on large image datasets due to the time it takes to analyse the images. The speeded-up robust features (SURF) algorithm, CNN or SIFT-CNN can be used to as an alternative to SIFT algorithm to decrease SoshaMapBot's time performance.

The speeded-up robust features (SURF) algorithm is similar to SIFT algorithm, but much faster and more robust for local, similarity invariant representation and comparison of image, its drawback though is that it is unstable to changes in rotation and illumination (Jain et al., 2017). The advantage of using SURF over the SIFT approach is its fast computation of operators using box filters that enable real-time applications such as tracking and object recognition (Bay et al., 2006), and it has also been proved to give good results outputs of image analysis (Khan et al., 2011).

The convolutional neural networks (CNN) is a classification algorithm which operates in a set of non-linear functions and consist of a number of layers Fischer et al. (2014). When compared to SIFT, CNN has been found to yield a better performance time. Zheng et al. (2018) reported that SIFT takes approximately 1-2 seconds to extract regions from a 640 x 480 image, and it takes 0,082 seconds and 0.347 seconds to iterate through networks of 224 x 224 and 1024 x 768 image. Additionally, Fischer et al. (2014) examined the performance CNN on matching tasks when compared to SIFT, they found that CNN performs better than SIFT when extracting descriptors to perform a matching task. Drawbacks of using CNN over SIFT is that is requires large amount training datasets Fischer et al. (2014) and CNN has a weak performance on grey scaled images (Zheng et al., 2018).

SITF-CNN algorithm, proposed by (Tsourounis et al., 2022) is a an algorithm combining both the SIFT and CNN to form one algorithm. It works by feeding the SIFT image representation into CNN. The grey scale image remains an issue with SIFT-CNN algorithm as with CNN algorithm, and is highly suitable for small datasets (Tsourounis et al., 2022). The advantages of SIFT-CNN are that it has potential of eliminating the amount of training required by CNN algorithm by using the results from SIFT to train the model and use end-to-end learning scheme (Tsourounis et al., 2022).

From the above possible alternative algorithms provided, this study recommends the use of SIFT-CNN algorithm for future related work. This is because the SIFT-CNN algorithm has been proved to faster that both SIFT and CNN algorithms when implemented individually.

Conclusion

The focus of the research was the development of an innovative solution to support new students and visitors when searching for locations on a large campus. The SoshaMapBot solution is a chatbot that can identify the user's location by analysing an image uploaded to the chatbot using image processing. The chatbot identified a location where the user wants to go by analysing the text. The chatbot then directs the user with text to find the required location. This study adopted the design science methodology to design an innovative and usable chatbot artefact.

In order to predict the user's current location, feature matching was used in combination with OpenCV's SIFT method to match features of the image and predict their similarity.

Additionally, Gaussian elimination and Euclidean distance were used to determine the validity of the matched features. The chatbot uses the entity-recognition technique to process natural language and was built using the Microsoft Azure Bot Service.

The artefact developed proved itself to be usable and innovative as it injected a self-developed chatbot application to provide campus directions. The model was evaluated based on the F1 score and exhibited a score of 0.75, which is an acceptable score in the field of human-computer interaction (HCI) development.

For future development, the researcher recommends using the combination of SIFT and CNN (SIFT-ANN) algorithm instead of the SIFT algorithm to decrease the computation time, as the SIFT algorithm was slower when processing a large number of images. CNN algorithm is another alternative which can be used instead of SIFT algorithm, however, there are several factors to consider when choosing to use the CNN algorithm. Factors such as number of datasets, training time and complexity of the model. SURF can also be used as opposed to SIFT, however the rotation and illumination of the image's needs be considered as SURF is unstable to rotation and illumination changes.

References

Adamopoulou, E., & Moussiades, L. (2020). An Overview of Chatbot Technology. In I. Maglogiannis, L. Iliadis, & E. Pimenidis, *Artificial Intelligence Applications and Innovations* Cham.

Arthur, R. M., Humburg, P. J., Hoogenboom, J., Baiker, M., Taylor, M. C., & de Bruin, K. G. (2017, 2017/08/01/). An image-processing methodology for extracting bloodstain pattern features. *Forensic Science International, 277*, 122-132. https://doi.org/https://doi.org/10.1016/j.forsciint.2017.05.022

Bay, H., Tuytelaars, T., & Gool, L. V. (2006). Surf: Speeded up robust features. European conference on computer vision.

Cannas, V. G., & Gosling, J. (2021, 2021/11/01/). A decade of engineering-to-order (2010–2020): Progress and emerging themes. *International Journal of Production Economics, 241*, 108274. https://doi.org/https://doi.org/10.1016/j.ijpe.2021.108274

Dechert, M., Schöning, J., & Barkowsky, T. (2019). *Implementation and Evaluation of a Chatbot to Crowdsource Geotagged Images to Detect Mosquito Breeding*. University of Cambridge.

Fischer, P., Dosovitskiy, A., & Brox, T. (2014). Descriptor matching with convolutional neural networks: a comparison to sift. *arXiv preprint arXiv:1405.5769*.

Harrison, O. (2018, 10 September 2018). *Machine Learning Basics with the K-Nearest Neighbors Algorithm*. Retrieved 19 November 2022 from https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design Science in Information Systems Research. *MIS Quarterly, 28*(1), 75-105. https://doi.org/10.2307/25148625

Jain, S., Kumar, B. S., & Shettigar, R. (2017). Comparative study on SIFT and SURF face feature descriptors. 2017 International Conference on Inventive Communication and Computational Technologies (ICICCT).

Jansen van Vuuren, J. C., Grobler, M., Leenen, L., Chan, P., & Dawood, Z. (2016). Morphological Ontology Design Engineering: A Methodology to Model Ill-Structured Problems. In (pp. 262-291). https://doi.org/10.4018/978-1-5225-0007-0.ch014

Khan, N., McCane, B., & Wyvill, G. (2011). *SIFT and SURF Performance Evaluation against Various Image Deformations on Benchmark Dataset*. https://doi.org/10.1109/DICTA.2011.90

Lee, J., Jang, D., & Lee, J. (2019). Automatic Template-style Social Magazine Generation Service and Video Content Synthesis. 2019 International Conference on Information and Communication Technology Convergence (ICTC).

Li, Y., Yao, T., Hu, R., Mei, T., & Rui, Y. (2016). Video ChatBot: Triggering Live Social Interactions by Automatic Video Commenting. 15-19.

Pardasani, A., Sharma, A. K., Banerjee, S., Garg, V., & Roy, D. S. (2018). Enhancing the Ability to Communicate by Synthesizing American Sign Language using Image Recognition in A Chatbot for Differently Abled. 2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO).

Peras, D. (2018). CHATBOT EVALUATION METRICS: REVIEW PAPER. 36th International Scientific Conference on Economic and Social Development – "Building Resilient Society".

Prasomphan, S. (2019). Using Chatbot in Trading System for Small and Medium Enterprise (SMEs) by Convolution Neural Network Technique.

Raschka, S. (2018). *Machine Learning Lecture Notes*. University of Wisconsin-Madison. https://pages.stat.wisc.edu/~sraschka/teaching/stat479-fs2018/

Roelke, R. (2013). *CS 143: Local Feature Matching*. Brown University. Retrieved 19 November 2022 from https://cs.brown.edu/courses/cs143/2013/results/proj2/rroelke/#:~:text=Feature%20matching%20refers%20to%20the,set%20of%20distinctive%20key%2Dpoints

Singh, A., Ramasubramanian, K., & Shivam, S. (2019). Chatbot Development Essentials. In *Building an Enterprise Chatbot: Work with Protected Enterprise Data Using Open Source Frameworks* (pp. 35-53). Apress. https://doi.org/10.1007/978-1-4842-5034-1_3

Tsourounis, D., Kastaniotis, D., Theoharatos, C., Kazantzidis, A., & Economou, G. (2022). SIFT-CNN: When Convolutional Neural Networks Meet Dense SIFT Descriptors for Image and Sequence Classification. *Journal of Imaging, 8*(10).

Zheng, L., Yang, Y., & Tian, Q. (2018). SIFT Meets CNN: A Decade Survey of Instance Retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 40*(5), 1224-1244. https://doi.org/10.1109/TPAMI.2017.2709749