International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

VRSS: A Touch-to-Vision-Text-Audio Artificial Multi-Modal Sensory System to Demonstrate Neural Network Processes

Vijay Harkare*[1], Riddhi Sanghani[2], Shruti Prasad[3], Sanika Ardekar[4], Dr. Prof. Aruna Gawade[5], Dr. Prof. Ramchandra Mangrulkar[6]

Submitted: 24/04/2023 **Revised**: 26/06/2023 **Accepted**: 06/07/2023

Abstract*:* The human brain is the most complicated human organ, and simulating its functionality is an exceedingly challenging task, particularly the multi-modal sensory functionalities of the brain. Results from biological experiments show that it is possible to identify instances of objects using tactile signals. This research uses similar concepts for modelling a multi-modal sensory input processing system for tactile inputs. VRSS is a novel touch-to-vision-to-text-to-audio system which simulates the multi-modal sensory behavior of the brain by converting tactile inputs to visual images, which are further converted to audio and text. The main aim of this research is to classify object instances based on tactile signals. Tactile inputs are captured and implicitly converted to visual inputs using the DIGIT sensor simulated in the TACTO simulator, and using them, the object is classified using Convolutional Neural Networks. The classification output is further converted into audio, thus successfully simulating three modalities - touch, vision, and sound. For construction of VRSS, multiple pretrained CNNs with different configurations of hyperparameters were tested, and the pretrained ConvNeXtTiny model had the best accuracy of them all - 91%. It was further modified, and the accuracy of the resulting custom VRSS CNN Model was found to be 95.83%. Following these results, this research will help in expanding the applicability of different CNNs. Along with this, it will also facilitate in-depth understanding of the human multi-modal sensory system, and also has wide scope in the fields of artificial intelligence and robotics, particularly in the navigation of uncharted territories.

Keywords: *Classification, Convolutional Neural Network (CNN), DIGIT sensor, multi-modal, touch, vision.*

1. Introduction

The 5 sensory organs present in the human body allow humans to accurately perceive the world as it is. They also capture crucial information about various events happening in the environment. The human brain perceives the object through these sensory organs and one of them is the skin, which provides the sensation of touch. In robots, this sense of touch can be replicated with the help of tactile sensors. The human brain also employs shared models of objects across multiple sensory modalities such as vision and tactile sensing so that knowledge can be transferred from one to another [1].

1 Student, Department of Computer Engineering, Dwarkadas J. Sanghvi College of Engineering, Mumbai – 400 056, INDIA
ORCID ID : 0009-0002-8671-4786
2 Student, Department of Computer Engineering, Dwarkadas J. Sanghvi College of Engineering, Mumbai – 400 056, INDIA
ORCID ID : 0009-0007-6783-4340
3 Student, Department of Computer Engineering, Dwarkadas J. Sanghvi College of Engineering, Mumbai – 400 056, INDIA
ORCID ID : 0009-0002-8011-3883
4 Student, Department of Computer Engineering, Dwarkadas J. Sanghvi College of Engineering, Mumbai – 400 056, INDIA
ORCID ID : 0009-0007-6618-0218
5 Assistant Professor, Department of Computer Engineering, Dwarkadas J. Sanghvi College of Engineering, Mumbai – 400 056, INDIA
ORCID ID : 0000-0001-8941-5381
6 Associate Professor, Department of Computer Engineering, Dwarkadas J. Sanghvi College of Engineering, Mumbai – 400 056, INDIA
ORCID ID : 0000-0002-9020-0713
** Corresponding Author Email: vijay.harkare2020@gmail.com*

The system VRSS presented in this research can be used to understand the working of the brain for object recognition using the sense of touch with the help of artificial model. It helps to increase the dynamics of robotic vision. By analyzing the data obtained from the sensors, a robot can recognize the properties of an object and determine its shape, size, texture, and other relevant information. In recent years, the rapid advancement of tactile devices and skins has opened new possibilities for incorporating tactile sensing into diverse robotic applications [2]. This will also facilitate research in the robotics field by helping to understand the relationship between the senses of touch and vision.
Various models are tried and studied to understand the relation between the sense of touch and the sense of vision. The essence of cross-modal retrieval research lies in the ability to learn a shared subspace where items from different modalities can be directly compared to one another [3]. The senses of touch and vision are complementary and have often been combined to form a multi-modal perception framework for understanding object geometry [4].
Using the outcome of the comparison of models, fine-tuning was done on the best model to build the custom model which has better accuracy. The custom model can further be used by the research community to understand the touch-to-vision transformation with better accuracy.
In the next section, an overview of the different approaches that have been previously applied to tackle this challenge of converting tactile inputs to visual outputs is presented.

2. Literature Survey

Previous work on the topic of converting touch to vision includes a few notable and relevant implementations.

The MIT paper [5] that examines the mechanics of human object grasping will supplement vision-based robotic object handling. It demonstrates how sensors evenly dispersed over the hand may be utilized to detect specific objects, estimate their weight, and investigate common tactile patterns using a scalable tactile glove and deep convolutional neural networks.

Different measures such as using pressure sensors and measuring both touch and orientation are taken into consideration in T-SGs and GT-SGs respectively [6]. The challenge arises that designing such high-architecture gloves is easy, but its implementation is difficult.

The user's palm housed a cluster of fifteen Force Sensitive Resistors (FSRs), which were used to determine the region of interaction in [7]. The active zones at the palm during contact with each surface at various forces were demonstrated by experimental data. The overall recognition rate is 84 percent, indicating that the user can confidently recognize four patterns.

Utilizing deep neural networks, active sensing, and learning to address the issue of robotic visuotactile cross-modal object recognition was demonstrated in [8]. The proposed work does unsupervised cross-modal transfer learning and actively learns from tagged visual point cloud examples. Additionally, an active tactile object recognition approach is used in the suggested work.

By modelling the relationship between the retrieved features and the human tactile sensation scores gathered via sensory evaluation tests, Ito et al. created a machine-learning model that assesses the tactile feeling of an unknown sample [9]. The gaps in this paper are that more diverse shapes and samples of different materials could not be identified properly. This research paper provides a method for detecting the objects sensed by using an encoder-decoder architecture.

A new model known as ResNet10-v1 was proposed in a different study [10] by fusing a convolutional neural network and a residual network. It enhanced the model's convolutional kernel, hyperparameters, and loss function, and used the K-means clustering technique to increase the accuracy of target categorization further. Through a significant number of trials, the study proved the viability and efficacy of the suggested strategy.

For the creation of cross-modal visual-tactile data, a residue-fusion GAN trained with additional feature-matching and perceptual losses is suggested by Cai and colleagues [11]. The paper uses the tactile data—pen sliding motion on the surface—as well as the visual image of a material surface as the visual and tactile data. The paper's findings demonstrate that the model performs significantly better than the baseline model in both the visual and tactile domains for recognition.

Deep CNN architectures have been employed to test the feasibility of learning transfer from vision to touch, confirming that visual and tactile data overlap some properties at some levels [12]. The topic was examined using various types of touch sensors in the article.

A unique framework for the creation of cross-modal sensory data for tactile and visual perception was suggested in [13]. Using the sense of texture as an example, conditional generative adversarial networks were used to produce fictitious tactile or visual outputs using input from the other modality. However, quality of image generation process can be improved and training the network on more datasets could increase the capabilities of the model.

Given a visual and tactile observation, Lin et al. describe a multi-model instance recognition system that can determine whether or not these observations pertain to the same object [14]. Here, a dataset of tactile observations and photographs is gathered using a robot outfitted with two GelSight touch sensors, one on each finger, and a self-supervised, autonomous data collecting technique.

Usage of conditional adversarial networks to suggest links between vision and touch was presented in [15]. According to the paradigm, findings for cross-modal prediction of known and unseen items are expected to be positive [15]. To link vision and touch, researchers have created novel activities that involve creating convincing tactile signals from visual inputs and thinking how to interact with objects when tactile data is provided as input.

Li and colleagues, in [16], focus on the cross-modal visual-tactile transformation based on deep learning. The classification information for the image is first obtained using the Resnet, then the spectrogram generator G and classification data from the most recent output are combined with G using ensembled GANs. The experimental findings demonstrate that the model can convert the texture image's visual data into tactile data. The tactile feedback of the item material is, however, restricted by the visual classification effect and the visual inaccuracy will carry over into the tactile feedback in the real tactile test.

Demonstration of object recognition from Piezoresistive tactile sensors has also been evaluated in [17]. The method implemented uses a tactile sensors' array of 16 rows × 16 columns matrix for taking tactile input. The tactile data from sensors is processed for resolution enhancement. Further, several DCNN models were implemented to get the best model. However, the paper suggests implementing multimodal learning and object exploration to increase recognition rate.

Falco and colleagues trained a classifier using visual data obtained from a Kinect camera, and objects are recognized at execution time solely based on tactile data, without any prior tactile information [18].

Deep convolutional neural networks, GAN and convolutional adversarial network have been primarily employed to derive a relationship between extracted features from images to human tactile sensation features. Sensors such as pressure sensors, force sensitive resistors, piezoresistive tactile sensors have been heavily used to locate the object on the hand. Accordingly, this study first highlights the proposed model to satisfy the aim of the research, followed by a discussion on the various models, technologies and components used in this research. Following this, the experimentation undertaken, which includes the preliminary work undertaken to understand the intricacies of visual and tactile data collection, manipulation, conversion and visualization, along with the preliminary design of the VRSS Model (The Hand Model), a description of the dataset used, the comparison between various designing strategies, simulated hardware components like sensors, different electrical phenomena that could be leveraged, and different machine learning models and their relative setups has been presented. After experimentation, a summary of the results of the experimentation, the corresponding discussion on those results, and finally the derived conclusion along with the future scope of this research work is presented.

3. Proposed VRSS Model

Various models and configurations have been experimented with in this research work, such as the Mathematical Model, the Hand Model and the Palm Model, which have been explained in detail in the subsequent sections. Out of those models, the best model obtained was the Palm Model. Hence, the Palm Model has been selected as the proposed model, since it satisfies the aim of this research work considering various real-world constraints.

The main aim of the Palm Model is to mimic the somatosensory functionalities of the human palm. The model will simulate the human palm using the simulated DIGIT sensor, collect tactile data, transform it to visual data, and classify the object placed on the palm of the hand. It is explained in detail in the following sub-section.

3.1. The Palm Model

The Palm Model has its basis in the fact that the palm of the hand has the highest sensing area of the entire available sensing area on the hand. Therefore, to successfully classify the object within a certain level of accuracy, it is reasonable to use just the palm of the hand to sense the object. In this model, the simulation contains just one DIGIT sensor (whose sensing area represents the palm of the hand), which senses the shape and size of the object, classifies the object and finally provides the output in audio and text format.

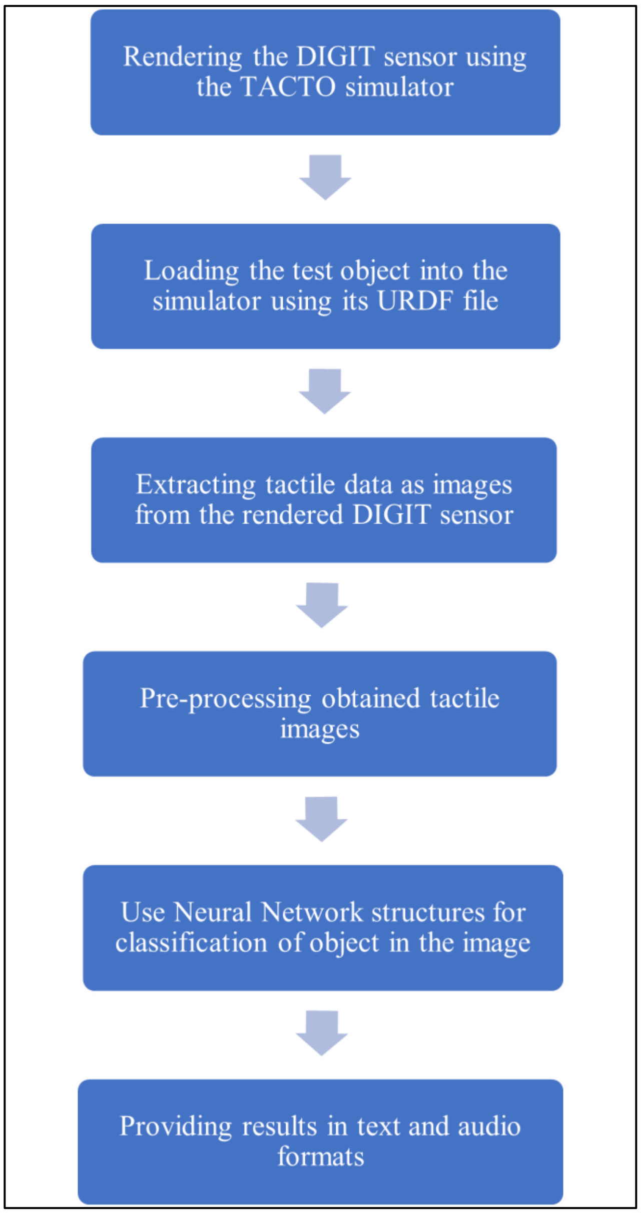

Rendering the DIGIT sensor using the TACTO simulator
Loading the test object into the simulator using its URDF file
Extracting tactile data as images from the rendered DIGIT sensor
Pre-processing obtained tactile images
Use Neural Network structures for classification of object in the image
Providing results in text and audio formats

Fig. 1. Flow of Palm Model

The various stages involved are depicted in Fig. 1 and are described below:

3.1.1. Rendering the DIGIT sensor using the TACTO simulator

Using the open-source configuration files and the 3D model files of the DIGIT sensor, a 3D model of the DIGIT sensor (along with sensing surface) is rendered in the TACTO simulator.

3.1.2. Loading the test object into the simulator using its URDF file

To obtain tactile images from objects in the simulator, the 3D model file of the object needs to be created using CAD software like Microsoft 3D Builder and Blender. These object files (which are in the form of waveforms or point-clouds) need to be loaded into the simulator using Unified Robot Description Format (URDF) files, which define the physical properties of the object such as its size, inertia, mass, collision properties, etc.

3.1.3. Extracting tactile data as images from the rendered DIGIT sensor

Once the object is rendered in the simulator alongside the DIGIT sensor, the object is then moved towards the sensing surface of the DIGIT sensors using a control panel which allows translation as well as rotation of the object in 3 axes each (therefore 6 DOF are available for each object). Then, the object is brought in contact with the sensing surface, and subsequently tactile image is recorded.

3.1.4. Pre-processing obtained tactile images

Since simulated tactile images contain noise, pre-processing techniques need to be employed to improve the accuracy of the classification model.

3.1.5. Use Neural Network structures for classification of object in the image

The images obtained from the DIGIT sensor are noisy. Hence, to learn the implicit patterns present, and to classify the images efficiently and effectively, Deep Neural Network techniques have to be utilized for classification tasks.

3.1.6. Providing results in text and audio formats

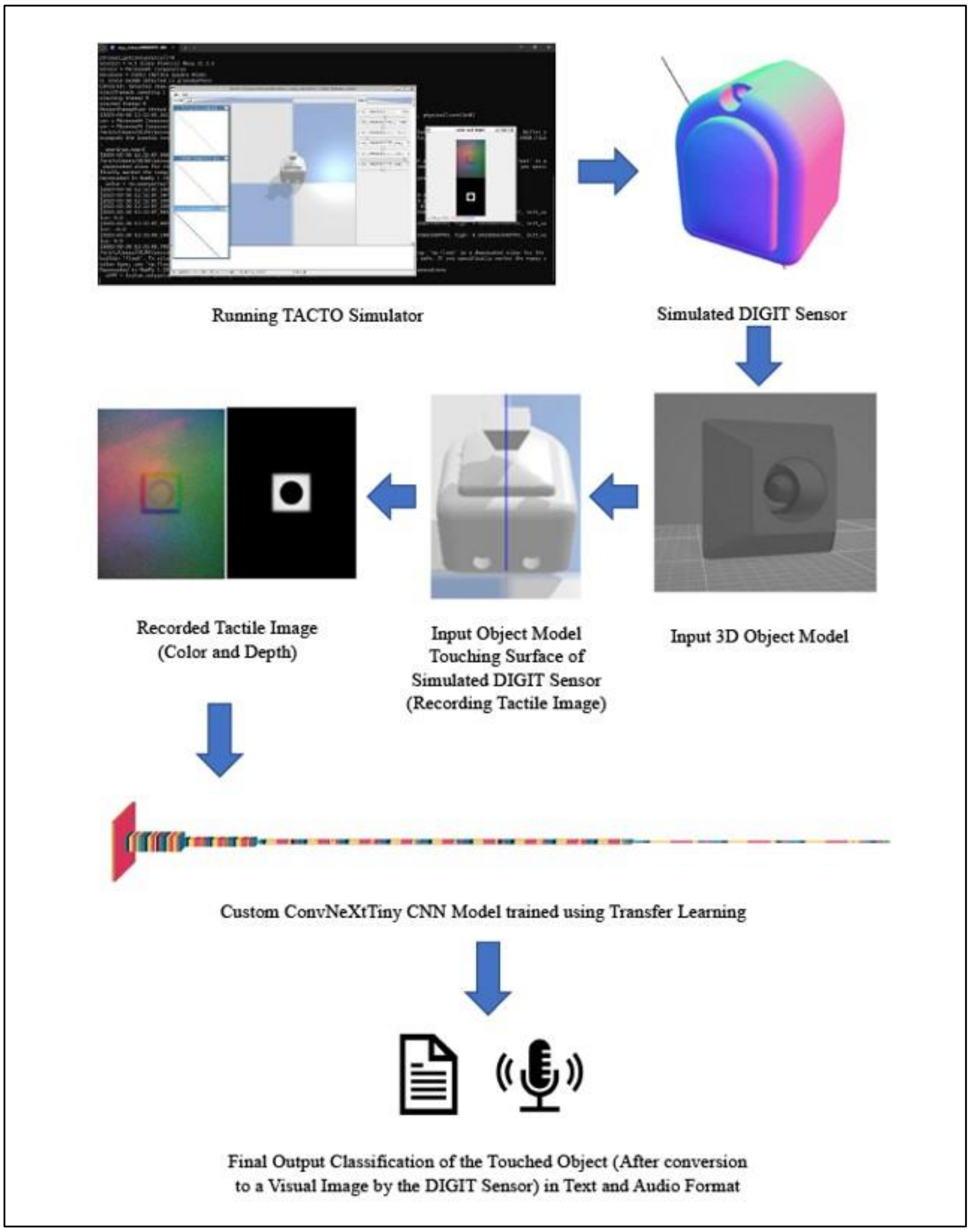
To increase the accessibility of the results for most of the audience, the results of classification are given both in textual and audio formats.

The Palm Model is fairly simple and straightforward and can be easily implemented considering real-world processing constraints. But it may fail to capture certain geometrical relationships between some parts of the objects, since only palm of the hand is touching the object. To improve upon this point, the Hand Model, which is explained in the Methodology and Experimentation section, can be used if enough resources are available (since the Hand Model was proven to be extremely resource extensive.)

The complete architecture of the Palm Model (which is also referred to as the VRSS Model in this research) is provided in the next section.

4. Model Architecture

The architecture diagram in Fig. 2 depicts, at a higher level, all the components discussed in the section on Proposed Model, and the various transitions associated.



Running TACTO Simulator
Simulated DIGIT Sensor
Recorded Tactile Image (Color and Depth)
Input Object Model Touching Surface of Simulated DIGIT Sensor (Recording Tactile Image)
Input 3D Object Model
Custom ConvNeXtTiny CNN Model trained using Transfer Learning
Final Output Classification of the Touched Object (After conversion to a Visual Image by the DIGIT Sensor) in Text and Audio Format

Fig. 2. Model Architecture Diagram

In the next section, the various components and models that have been experimented with in this research have been discussed.

5. Components and Models

To understand the following research, it is important to understand the various components and models involved. These include the DIGIT sensor, TACTO simulator, Xception, EfficientNetV2, ResNet152V2, ConvNeXtTiny, VGG-16 and InceptionV3.

5.1. DIGIT Sensor

It is a low-cost, small, and high-resolution tactile sensor designed for in-hand handling. Many tactile sensors measure force at a single point or patch of contact, providing potentially rich dynamic information but limited geometric information [19]. DIGIT improves upon past vision-based tactile sensors by miniaturizing the form factor to be mountable on multi-fingered hands, and by providing several design improvements that result in an easier, more repeatable manufacturing process, and enhanced reliability [20]. In this study, tactile input is gathered using a set of emulated DIGIT sensors.

5.2. TACTO Simulator

TACTO is a quick, adaptable, and free simulator for tactile vision sensors. TACTO produces high-resolution and high-fidelity reading from tactile sensors at high frequency (>100 Hz) [21]. The DIGIT and OmniTact tactile vision sensors may be easily simulated using this simulator, which can produce realistic high-resolution touch readings at hundreds of frames per second. The simulator has been utilised for this research project to mimic various biological sensing configurations.

5.3. Xception

The Keras deep learning library provides an implementation of the Xception model, which is pre-trained on the ImageNet dataset. This model is available in the Keras applications module and can be easily loaded using the Keras API. The Xception model in Keras has 88 convolutional layers and achieves state-of-the-art performance on the ImageNet dataset. In order to reduce the number of parameters and increase computational performance, the model makes use of depth wise separable convolutions.

5.4. EfficientNetV2

The Keras deep learning library provides an implementation of the EfficientNetV2 model, which is a family of convolutional neural network architectures that were introduced in 2021 as an improvement over the original EfficientNet models. EfficientNet properly adjusts depth, width, and resolution when the network is shrunk, outperforming other cutting-edge algorithms in effectiveness [22]. These models are designed to be both efficient and effective for a wide range of computer vision tasks. Once the model is loaded, it can be used for image classification or as a feature extractor for transfer learning. EfficientNetV2 models achieve state-of-the-art performance on several computer vision benchmarks while having fewer parameters and requiring less computational resources compared to other models. This makes them a popular choice for a wide range of computer vision tasks.

5.5. ResNet152V2

The Keras deep learning library provides an implementation of the ResNet152V2 model, which is a variant of the ResNet model that has 152 layers. ResNet152V2 is a very deep neural network architecture that has achieved state-of-the-art performance on several computer vision benchmarks. Once the model is loaded, it can be used for image classification or as a feature extractor for transfer learning. ResNet152V2 uses residual connections to mitigate the vanishing gradient problem and has achieved state-of-the-art performance on several computer vision benchmarks. It is a popular choice for a wide range of computer vision tasks, including image classification, object detection, and semantic segmentation.

5.6. ConvNeXtTiny

ConvNeXtTiny is a CNN of the family ConvNeXt. ConvNeXt is a family of convolutional neural network architectures that were introduced in 2018 and are known for their ability to achieve high accuracy with a relatively small number of parameters. The ConvNeXtTiny model can be used with pre-trained weights or can be trained from scratch.

5.7. VGG-16

A deep convolutional neural network that has shown remarkable success in image classification is the Keras VGG-16 model. It starts with a string of convolutional layers with tiny 3x3 filters, then moves on to layers with maximum pooling for down sampling. Its uniform design with several stacked convolutional layers, which makes it deeper than many other models at the time of its inception, is the distinctive feature of VGG-16. The model can learn complicated characteristics and recognise minute details in photos because of its deep architecture. There are 16 weight layers in the VGG-16 algorithm, including convolutional, pooling, and fully connected layers. It has had prior training on big picture datasets like ImageNet, where it has displayed outstanding ability in categorising images. Pretrained weights are readily available, which enables effective transfer learning, where the model is modified to fit new tasks and datasets by adjusting the pre-learned representations. For computer vision applications, the VGG-16 model in Keras offers a dependable and popular architecture that strikes a compromise between performance and ease of use.

5.8. InceptionV3

The InceptionV3 model in Keras is a powerful deep neural network architecture that facilitates visual feature extraction and understanding. With its pretrained weights, it offers a convenient solution for transfer learning in various computer vision tasks, and its integration with Keras makes it easy to incorporate it into deep learning projects. InceptionV3 in Keras comes with pretrained weights that have been trained on the ImageNet dataset. These pretrained weights capture generalizable visual features learned from a large number of images and can be leveraged for transfer learning. InceptionV3 can be used for a wide range of computer vision tasks, including image classification, object detection, and image segmentation. Its deep architecture and sophisticated feature extraction capabilities make it suitable for tasks that require understanding complex visual patterns and structures.

5.9. pyttsx3

pyttsx3 is a text-to-speech conversion library in Python. It is an offline, cross-platform solution that works with both Python 2 and 3. It supports multiple text-to-speech engines, including SAPI5 on Windows, NSSpeechSynthesizer on macOS, and eSpeak on Linux. It allows you to control voice properties such as rate, volume, and voice and also supports event callbacks to track the progress of the speech.

5.10. gTTS

gTTS (Google Text-to-Speech) is a Python library and command-line tool that interfaces with Google Translate’s text-to-speech API. It allows you to input text and convert it into spoken audio. The audio data can be saved to a file or played directly. gTTS supports multiple languages and dialects and allows you to control the speed of the speech. However, it requires an internet connection to work since it relies on Google’s servers to generate the audio.

This completes the discussion on the various components and models used in this research. The next section outlines the research methodologies applied and the experimentation carried out for this research work on the VRSS Model.

6. Methodology and Experimentation

To clearly understand the reason behind selecting the Palm Model as the proposed model, it is crucial to first understand the preliminary work undertaken before designing the Palm Model. This work is described in the following sub-section.

6.1. Preliminary Work

The preliminary work for this research consists of the Mathematical Model and the Hand Model. Both of these models are explained subsequently.

6.1.1. Basic Mathematical Model

This model represents the first iteration of the research work. Here, only a single DIGIT sensor simulated using TACTO simulator is used to record the tactile input from small-scaled cuboidal objects, and a simple mathematical model coupled with a computer vision model (to analyse and interpret the pre-processed tactile image obtained) is used to estimate the dimensions of the object, which are subsequently used to reconstruct a 3D model of the same object.

The reconstructed model is shown in Fig. 3.

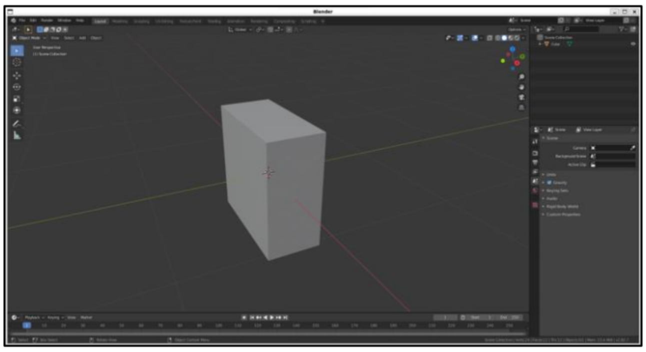


Fig. 3. Reconstructed 3D Model

Although this model is extremely simple and fast (as it depends on basic mathematics), it has many flaws. Firstly, it fails to capture the shape-related information of objects that do not conform to any standard geometric shape. Secondly, as the number of faces in the object increases, the complexity of the model increases steeply. This is clear from the graph in Fig. 4.

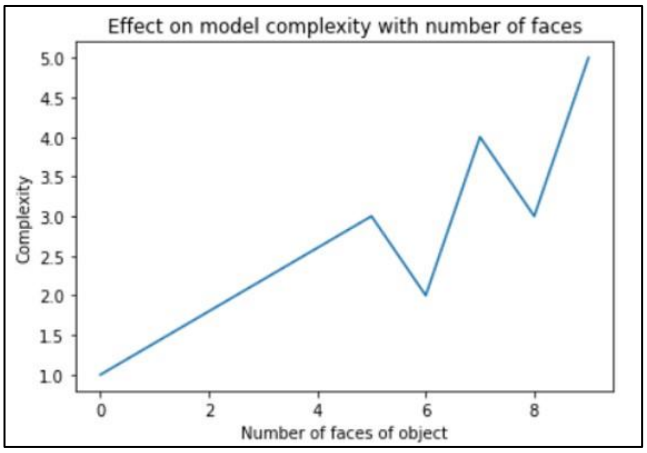


Fig. 4. Relation between Complexity and number of faces of object.

Graph in Fig. 5 is based on the Table 1. Table 1 demonstrates the increasing complexity of the shape prediction with increasing number of faces. The complexity of the object prediction is represented using a numerical range of 1-5 where 1 stands for lowest complexity in shape prediction and 10 stands for highest complexity. The objects with different number of faces require different number of faces to be sensed by the sensor to plot the 3D model. The number of faces sensed depends on the dimensions required to plot the 3D model. On finding the number of faces sensed, the calculations involved in predicting the dimensions is different for different shapes, which serves as a deciding factor for the complexity of the model.

Table 1. Relationship between number of faces in object, number of faces sensed and complexity

Number of faces in object	Number of faces sensed	Examples	Complexity
0	1	Sphere	1
5	2	Square Pyramid	3
6	2	Cube, Cuboid	2
7	2	Heptahedron	4
8	2	Octahedron	3
9	9	Enneahedron	5

These flaws led to the necessity of designing alternative models. One of those models was the Hand Model, which is explained next.

6.1.2. The Hand Model

As seen from the preliminary work on the mathematical model, a single DIGIT sensor can capture extremely valuable information regarding the shape and size of the object touching its surface. Keeping this property in mind, the next iteration of the research work was designed. It was called "The Hand Model". It is explained in the following text.

Early researchers extensively employed methods based on contact points due to the low resolution of tactile sensors and the prevalence of single-contact force sensors. Allen et al. [23] proposed fitting a super-quadric surface to sparse finger-object contact points and utilized the recovered super-quadric parameters for haptic object recognition. Similarly, in [24], a polyhedral model was derived by leveraging the locations of contact points and hand pose configurations to reconstruct the shape of unknown objects. Pezzementi et al., in [25], introduced a method for constructing an object representation by mosaicing tactile measurements. Meier et al. [26] utilized Kalman filters to generate 3D representations of unknown objects using contact point clouds collected by tactile sensors and applied the ICP algorithm for object classification. In [27], contact point clouds were combined with voxel representations to model object shapes. These methods enabled the retrieval of arbitrary contact shapes, but they can be time-consuming for investigating large object surfaces due to the requirement of excessive contacts.

The Hand Model, on the other hand, is based on the skeletal structure of the human hand. A total of 23 simulated DIGIT sensors are arranged in the shape of a human hand and are placed at strategically important points on the hand. When the rendered object is brought in contact with the sensor, array, sensors at the point of contact are activated, and tactile inputs from all sensors are collected. These inputs are then arranged in a 4 * 6 grid of tactile images (where some cells are left blank due to the absence of sensors in the corresponding positions), and a single image is used for further tasks. This image is then used to classify the object, the result of which is provided in textual and audio format.

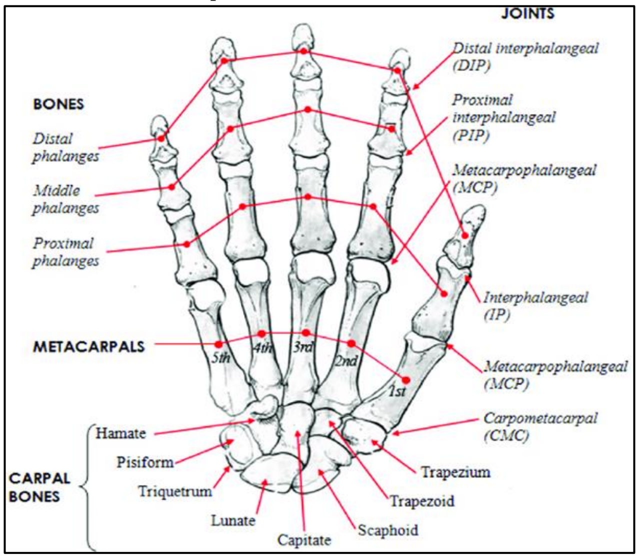


Fig. 5. Human hand skeletal structure depicting finger bones, joints, metacarpals, and carpal bones [28]

Understanding the skeletal structure of the hand is extremely important to determine the arrangement of sensors in the simulated sensory hand. As can be clearly seen from Fig. 5, each finger except the thumb has 2 joints on in the finger, and one base joint (which joins the finger to the palm, or with respect to the skeletal structure, to the Metacarpals and Carpal Bones). And for the thumb, there is one joint in the finger, and one base joint. Thus, from this skeletal structure it can be concluded that, to accurately capture sensory and geometric information as captured by the human hand, it is beneficial to have 3 sensors (out of which one is at the fingertip) on the 4 fingers and 2 sensors on the thumb. Contact sensing can typically be achieved with force/torque sensors, joint torque sensors and pressure sensor arrays as well [29]. But, to increase the quality of tactile data captured, and to facilitate implicit touch-to-vision conversion, this model makes use of the simulated DIGIT sensors. The region covered by the Metacarpals and the carpal bones (generally known as the palm) can be modelled using 9 sensors arranged in a 3 * 3 grid of sensors. Thus, a total of 23 simulated DIGIT sensors will be rendered in the TACTO simulator.

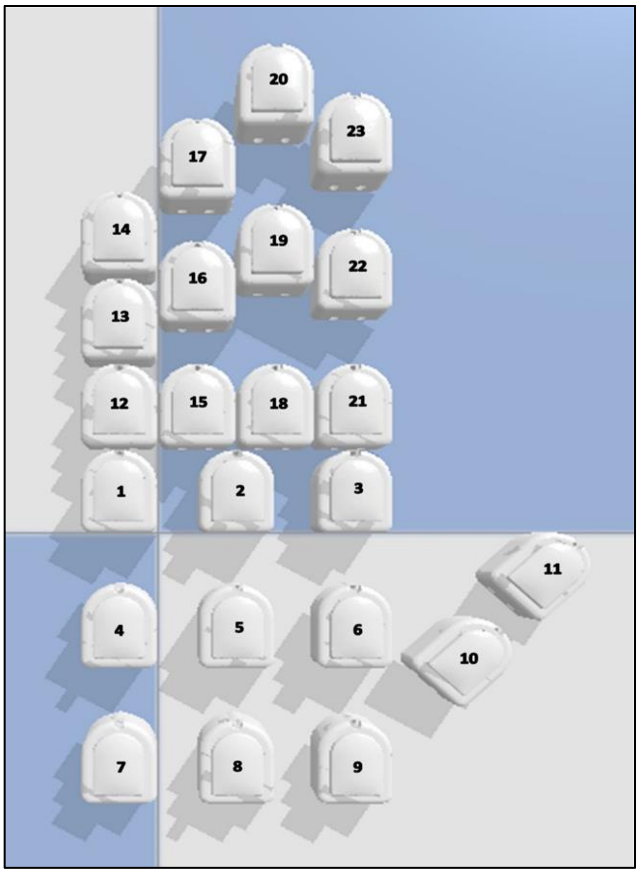


Fig. 6. Simulated Hand in TACTO Simulator

The corresponding simulated hand in the TACTO simulator is as shown in Fig. 6.

The sensors are numbered to keep track of corresponding tactile images in the 2D image array (which will be useful to involve the relative location of the sensors in classification task).

The tactile input obtained from such a simulated hand is of the form as shown in Fig. 7.

The main limitation of this model is the computational resources required. This model requires parallel processing of 23 optical tactile sensor inputs, which makes the data collection process very slow.

Now that the preliminary work and experimentation is clearly stated, in the following sub-subsections, the experimentation carried out for the VRSS Model (The Palm Model) is explained. This starts with the explanation of the dataset used, in the next sub-section.

6.2. Dataset

The dataset used for the experimentation is a hybrid dataset. The original dataset consists of a total of 1100 pressure images. These images have been collected from a high-resolution tactile sensor, which is basically an array of a number of pressure sensors. In the original paper, a dataset formed by tactile images has been collected. This dataset is used to train the TactNet models, and the classifiers of the transfer learning methods [30], in the original paper. The data from this dataset is expanded by adding the depth images collected from the simulated DIGIT sensor to fine-tune the dataset to make it usable with VRSS. The image-collection process is completely automated and collected images of the object in different configurations. Out of the 22 classes present in the original dataset, this research work uses only 6 out of those 22 classes, and one more new class has been added to the original dataset. The hybrid dataset contains images for 7 classes labelled as: ball, box, key, pliers, scissors, screw, and pen. The complete dataset contains 503 tactile images, which have been divided into training and testing sets.

The next sub-section describes the experimentation methods applied for the VRSS Model (The Palm Model).

6.3. VRSS Model Experimentation Description

As stated in the Proposed Model section, the Hand Model leads to excessive computational overhead, which is very large for even a powerful personal computer with good enough memory and GPU specifications. Therefore, hereafter, the Palm Model is referred to as the VRSS model.

As stated at the start of the Proposed Model section, the main aim of this research is to transform tactile signals to visual signals, and then to classify those visual signals. The methodology described below tries to achieve this aim.

As discussed in the section for proposed models, the Palm Model consists of a single DIGIT sensor representing the sensing area of a human palm. The setup is as shown in Fig. 8.

This setup is used to convert the tactile signals to visual signals. This is done implicitly by the simulated DIGIT sensor. The tactile image obtained from such a sensor is as shown in Fig. 9.

The obtained image corresponds to the visual signal corresponding to the tactile signal collected by placing the object on the sensor (which represents the palm of the human hand).

This image is then subjected to various preprocessing steps before it is given to the CNN model for classification. These preprocessing steps are given in the following text (The same preprocessing steps are also applied while training the CNN models).

6.3.1. Preprocessing of images

1. The images ae first loaded from either the training or the testing dataset, depending upon the task to be performed.
2. Next, the images are resized to a size of 224 × 224, which is required for the CNN models.
3. These images are then converted to NumPy arrays.
4. Finally, the dimensionality of the arrays from step 3 is altered by adding a new axis at the position 0 of those arrays.

This completes the preprocessing of the images that is required to

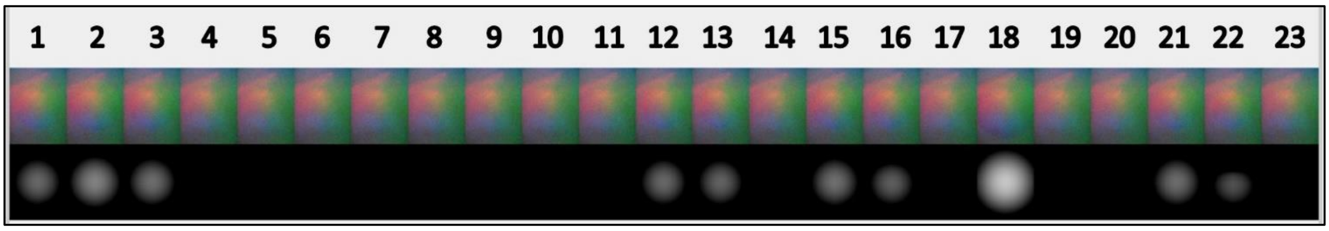


Fig. 7. Tactile Images from Sensors numbered 1 to 23

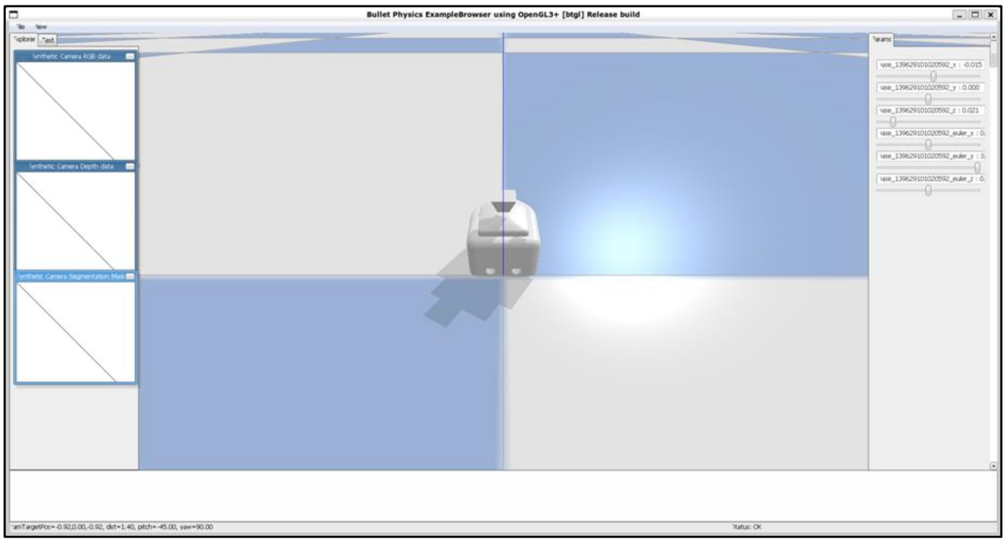

Bullet Physics ExampleBrowser using OpenGL3+ [btgl] Release build
Synthetic Camera RGB data
Synthetic Camera Depth data
Synthetic Camera Segmentation Mask
Params

Fig. 8. DIGIT Sensor in TACTO Simulator

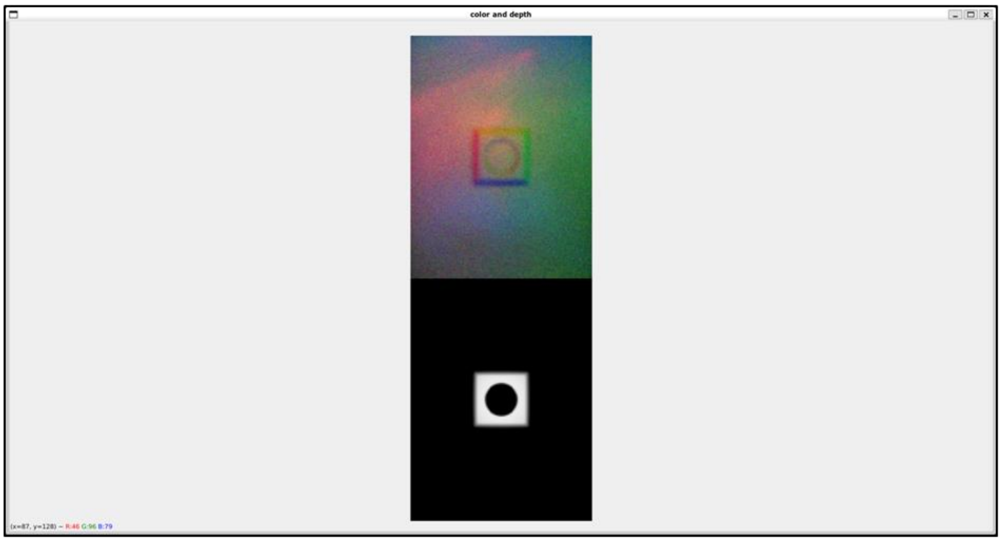


Fig. 9. Image obtained from simulated DIGIT sensor

be carried out before training and testing the CNN models.

The remaining sub-sections describe the process of experimenting with different CNN models, and how transfer learning is used for classification using these Keras models.

6.3.2. Use of Transfer Learning of Keras models for Classification

6.3.2.1. Keras Models

For classification, a comparison of six Keras models- Xception, ResNet152V2, EfficientNetV2, VGG16, InceptionV3 and ConvNeXtTiny was made. Keras models are convolutional neural networks. A pre-trained version of the network trained on more than a million images from the ImageNet database can be loaded. The pre-trained network can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals. As a result, the network has learned rich feature representations for a wide range of images. The network has an image input size of fixed size. Since the imagenet was not pre-trained on tactile images, transfer learning of Keras models was used.

Machine learning methods typically assume that the training and test data belong to the same feature space and distribution. However, when the distribution changes, it is expensive or impossible to collect new training data and rebuild the models. To address this challenge, transfer learning or knowledge transfer between task domains is utilized, minimizing the need for reacquiring training data [31], [32]. The intuition behind transfer learning for image classification is that if a model is trained on a large and general enough dataset, this model will effectively serve

Fig. 10. Custom VRSS CNN Model Architecture

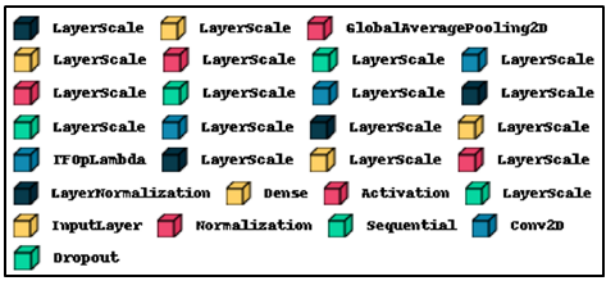


Fig. 11. Colour Index for layers in Custom VRSS CNN Model Architecture

as a generic model of the visual world. When strongly labelled target data is available, the model can further be fine-tuned using the labelled bounding boxes to improve both the recognition and the object localization [33]. One can then take advantage of these learned feature maps without having to start from scratch by training a large model on a large dataset.

All of the Keras models mentioned were then trained on the dataset which consisted of 7 objects – ball, box, key, pen, pliers, scissors and screw.

All of these six models were trained with three different configurations each. These were as follows:

1. Batch Size: 32; Learning Rate: 0.01; Epochs: 7
2. Batch Size: 32; Learning Rate: 0.05; Epochs: 10
3. Batch Size: 32; Learning Rate: 0.01; Epochs: 10

The best configuration out of these configurations (according to accuracy and loss) was then determined for each of the CNN models, after training and evaluating all of the six Keras models with these three configurations for each CNN model. The best model was then determined based on the accuracy and loss of these best-configuration-models (As is indicated in the Results and Discussion section later, this model was found out to be ConvNeXtTiny). This best model was further fine-tuned to build the final Custom VRSS CNN Model.

6.3.2.2. Custom VRSS CNN Model

The Custom VRSS CNN Model is based on the ConvNeXtTiny Model. Conditional adversarial networks, such as those used in ConvexNeXTiny show great promise in various image-to-image translation tasks, particularly when dealing with intricate graphical outputs that possess a high level of structure [34]. A two-step transfer learning process has been applied to improve the performance of the base pre-trained ConvNeXtTiny model. This approach involves adapting object models acquired in a specific visual domain to new imaging conditions. It achieves this by learning a transformation that minimizes the influence of changes in the feature distribution induced by different domains [35].

The model architecture can be visualized as shown in the Fig. 10.

The color index for the layers in the architecture is shown in the Fig. 11.

The first step is feature extraction, and the second step is fine-tuning.

Feature Extraction

In this step, all the classification layers after the flatten operation are removed. In this way, a network is obtained that is ideal for feature extraction. This is because the output features of the last layer before the flatten operation retain more generality as compared to the classification layers.

After this, the convolutional base so obtained is frozen before compiling and training the model. After freezing the weights of the base model, a new classification head consisting of a Global Average Pooling 2D layer, and a Dense layer is added.

The final model consists of a total of 27,825,511 parameters, out of which only 5,383 are trainable. Finally, the model is compiled using a categorical cross-entropy loss function with a learning rate of 0.001 for the feature extraction step. The trained model is saved for fine-tuning.

Fine-tuning

The previous model is further trained for fine-tuning the weights. In this model, the first 100 layers out of the 151 layers are frozen, while the rest of the layers are unfrozen. This is done because the lower layers learn very simple and general features while the higher layers learn more specialized features specific to the application, which in this case is learning features from tactile images.

The learning rate is kept at 0.0001, which is much smaller than the previous learning rate. This is because the aim here is to train a much larger model and readapt the pre-trained weights.

The number of trainable parameters in this model is 19,083,271. This is the final custom model which is trained on the training and validation images.

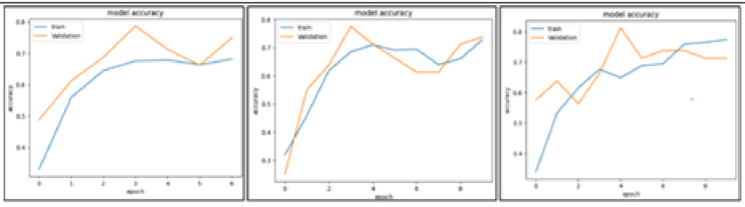
After the model gives its classification output, this output is preliminarily represented in the form of text, which limits the accessibility of the VRSS Model. Therefore, the next sub-section addresses the task of enhancing the accessibility of the VRSS Model by providing the classification output in the form of audio signals.

6.3.3. Providing Results in the form of Audio

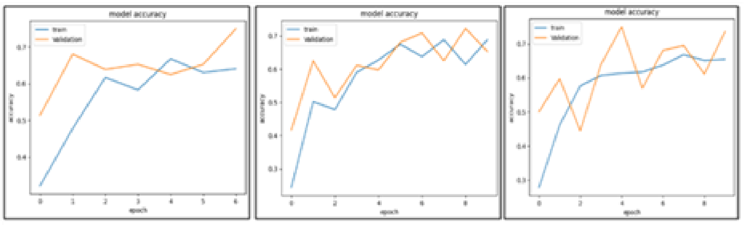
Various solutions exist which allow the conversion of textual data to audio data. Some of the most popular solutions to perform this conversion were found out to be the pyttsx3 library and the gTTS library, the details of which are provided in the Components and Models section covered previously.

As was indicated, pyttsx3 works seamlessly in offline mode as well, and supports various TTS engines, while the gTTS library only works in online mode. But gTTS has the added advantage of accessing the powerful Google Translate API.

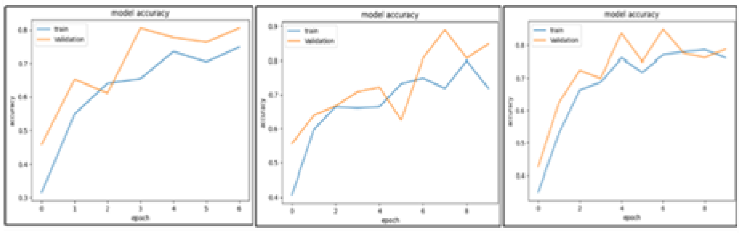
Hence, the VRSS model makes use of both these models. In the lack of a stable internet connection, the CRSS model utilizes the



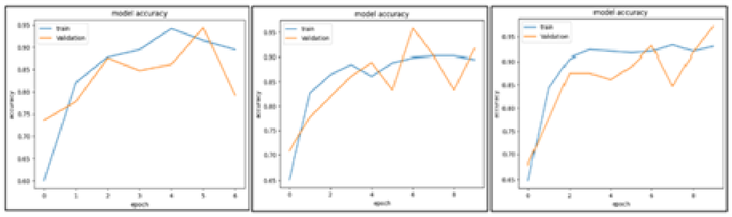
(a) Three configurations of Xception



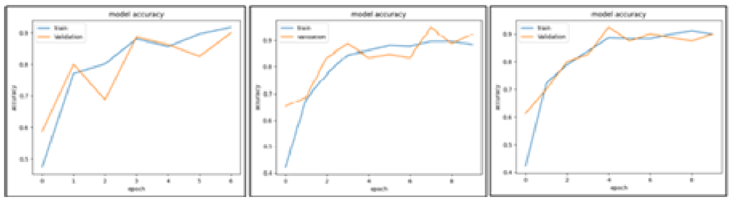
(b) Three configurations of InceptionV3



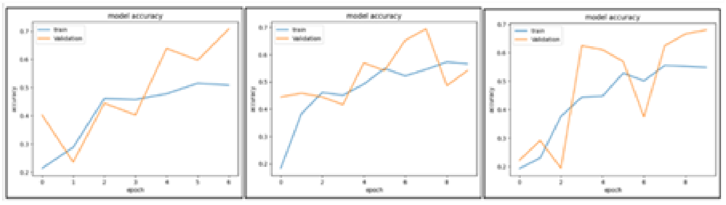
(c) Three configurations of VGG-16



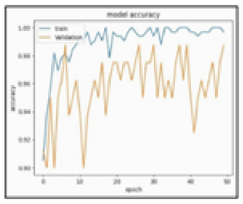
(d) Three configurations of EfficientNetV2



(e) Three configurations of ConvNeXtTiny



(f) Three configurations of ResNet152V2



(g) Custom VRSS CNN Model

Fig. 12. Three configurations each of the six models trained using transfer learning - Xception, InceptionV3, VGG-16, EfficientNetV2, ConvNeXtTiny, ResNet152V2 and Custom model

pyttsx3 library to perform the text-to-speech task, while in the presence of a stable internet connection, the model switches seamlessly to the gTTS library to perform the TTS task.
This completes the discussion of various methodologies applied in the experimentation phase of the VRSS Model. The next section highlights the key findings and results of this experimentation, and also provides discussion on the same.

7. Results and Discussion

As explained in the previous Methodology and Experimentation section, six different pre-trained Deep Learning models were used to classify the tactile image obtained from the sensor. The models used were imported from the Keras library. As explained in the Methodology and Experimentation section, each model was trained with three sets of different parameters and the best among them was selected to be customized for the dataset used in this paper. Epochs and Learning rate were the parameters which were fine-tuned to get the best results. Table 2 shows the best parameters and accuracy obtained by training them.

Table 2. Comparison of CNN Models

Model Name	Learning Rate	Number of epochs	Accuracy
Xception	0.01	7	0.843
EfficientNetV2	0.05	10	0.904
ResNet152V2	0.05	10	0.576
VGG-16	0.01	7	0.831
InceptionV3	0.05	10	0.843
ConvNeXtTiny	0.05	10	0.905

Fig. 12. shows the comparison between the model accuracy of different models (with different configurations of hyperparameters). The images for each model are ordered according to the parameters used to train those models. The first image corresponds to the model trained with the parameters [Batch Size: 32; Learning Rate: 0.01; Epochs: 7], the second one corresponds to the model trained with the parameters [Batch Size: 32; Learning Rate: 0.05; Epochs: 10] and the third one corresponds to the model trained with the parameters [Batch Size: 32; Learning Rate: 0.01; Epochs: 10].

Table 2 and Fig. 12 shows the comparison between the models which used parameters that gave the best output. Since the accuracy of the ConvNeXtTiny model was best, it was then modified to build the custom model which had higher accuracy and thus provided better results. The modifications made to the ConvNeXtTiny model were already explained in the previous section. But they are revisited in the next sub-section so that understanding the results of the model is easier.

7.1. Custom VRSS CNN Model

The custom model as mentioned in the methodology and experimentation section was trained on a total of 503 images, out of which 327 were used for training, 80 for validation and 96 for testing. The total number of classes was 7. The number of epochs for the feature extraction phase was set to 40, batch size to 32 and learning rate to 0.001, while for the fine-tuning phase, the number of epochs was 50, batch size was 32 and the learning rate was 0.0001. Comparing the weight updates in various convolutional layers reveals that a pre-trained CNN may effectively identify tactile data by updating a few of its convolutional layers on visual data.
After training the feature extractor, the training accuracy was obtained to be 83.75% with a validation loss of 0.5208.
After fine tuning the model, the validation accuracy increased to 94.99% and the loss decreased to 0.3549.
Also, after fine-tuning, the testing accuracy was found out to be 95.83% and the loss was found out to be 0.3058.
The accuracy and loss values of the final custom model were found out to be better than the pretrained models by a big margin. This was because, the weights of the specialist layers of the custom model, in addition to the new classification head, were fine-tuned.
The precision score of the Custom VRSS CNN Model was 0.9608, while the recall score of the model was 0.9583.
Fig. 12 depicts the confusion matrix obtained while evaluating the Custom VRSS CNN Model.

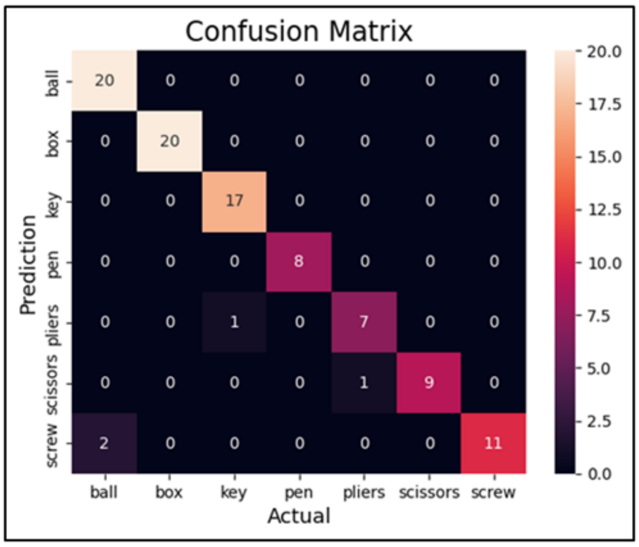


Fig. 13. Confusion Matrix for Custom VRSS CNN Model

The confusion matrix clearly shows the goodness of the model. This is further reinforced with the Table 3 given which contains the result given by the custom model after testing, for each object tested.

Table 3. Object classification statistics for Custom VRSS CNN Model

Object name	No. of object instances	No. of correctly identified object instances	No. of incorrectly identified object instances
Ball	22	20	2
Box	20	20	0
Key	18	17	1
Pen	8	8	0
Pliers	8	7	1
Scissors	9	9	0
Screw	11	11	0

Above results have been visualized as a bar graph as shown in Fig. 14, for better interpretability.

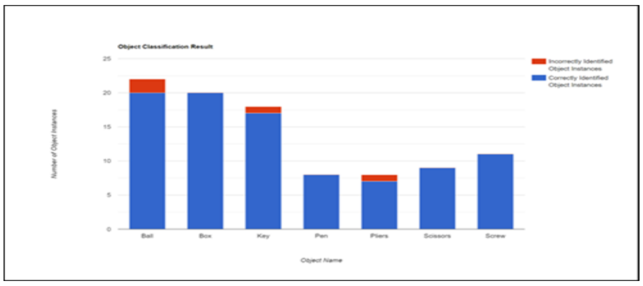


Fig. 14. Object Classification Results

This completes the discussion of the results obtained after performing the experimentation for the VRSS Model.

Finally, the conclusion of the entire research work conducted on the VRSS model is given in the next section.

8. Conclusion

In this research work, the working of the VRSS model which includes the conversion of tactile inputs to visual images whose result is further provided in the form of text and audio, has been highlighted. In this way, the presented model VRSS successfully models the multi-modal sensory neural processes. The tactile inputs are captured using DIGIT sensor. These tactile images are then fed into the Deep Convolutional Neural Networks (DCNNs) for object classification. The output is presented in the form of audio and text.

The paper compares various Deep Convolutional Neural Networks (DCNNs) such as XceptionNet, EfficientNet, ResNet152V2, VGG-16, InceptionV3 and ConvNeXtTiny to get the model with the best accuracy. The most accurate model comes out to be ConvNeXtTiny with an accuracy of 91.4%. The best accuracy model is further modified to improve the accuracy. The model on modification has a better accuracy of 95.83%. The model performs exceptionally well for predefined 7 objects; however, the number of objects can be further increased to make the model more diverse. DCNN fine-tuned transfer learning with Keras models technique which is used in building the custom model offers a powerful approach for touch to vision research. By adapting pre-trained models to learn the mapping between tactile and visual data, researchers can leverage existing visual knowledge, optimize the touch to vision mapping, and enhance the understanding and application of tactile information in various research domains.

The project successfully serves in predicting the objects from sensed data which contributes to the fields of robotics, AI and Deep Learning. The future scope of the project includes incorporating more objects to diversify the model. The future prospect of this research also includes the recreation of 3D model of the sensed object to facilitate object identification and perception in the fields of robotics, haptics, and medical sciences to provide relief for various vision and touch-related diseases.

Acknowledgements

This research was supported by the Department of Computer Engineering, Dwarkadas J. Sanghvi College of Engineering. We thank the HOD of the Department of Computer Engineering, Dr. Prof. Meera Narvekar for providing us with the opportunity to work on this research. We thank Dr. Prof. Aruna Gawade and Dr. Prof. Ramchandra Mangrulkar for their valuable insights in this research work, and also for helping in publishing this research work. Finally, we thank all our colleagues who guided us at various stages of this research work.

Author contributions

Vijay Harkare: Conceptualization of this study, Methodology, Dataset Preparation, Software, Validation, Investigation, Field Study, Writing - Original draft preparation **Riddhi Sanghani:** Conceptualization of this study, Methodology, Dataset Preparation, Software, Validation, Investigation, Field Study, Writing - Original draft preparation **Shruti Prasad:** Conceptualization of this study, Methodology, Dataset Preparation, Software, Validation, Investigation, Field Study, Writing - Original draft preparation **Sanika Ardekar:** Conceptualization of this study, Methodology, Dataset Preparation, Software, Validation, Investigation, Field Study, Writing - Original draft preparation **Dr. Prof. Aruna Gawade:** Conceptualization of this study, Reviewing - Original draft preparation **Dr. Prof. Ramchandra Mangrulkar:** Reviewing - Original draft preparation.

Conflicts of interest

The authors declare no conflicts of interest.

Data Availability

Data will be made available on request.

References

[1] S. Luo, W. Yuan, E. Adelson, A. G. Cohn, and R. Fuentes, "ViTac: Feature Sharing between Vision and Tactile Sensing for Cloth Texture Recognition," in Proceedings - IEEE International Conference on Robotics and Automation, 2018. doi: 10.1109/ICRA.2018.8460494.

[2] S. Luo, J. Bimbo, R. Dahiya, and H. Liu, "Robotic tactile perception of object properties: A review," Mechatronics, vol. 48. 2017. doi: 10.1016/j.mechatronics.2017.11.002.

[3] B. Wang, Y. Yang, X. Xu, A. Hanjalic, and H. T. Shen, "Adversarial cross-modal retrieval," in MM 2017 - Proceedings of the 2017 ACM Multimedia Conference, 2017. doi: 10.1145/3123266.3123326.

[4] [4] F. R. Hogan, M. Jenkin, S. Rezaei-Shoshtari, Y. Girdhar, D. Meger, and G. Dudek, "Seeing through your Skin: Recognizing objects with a novel visuotactile sensor," in Proceedings - 2021 IEEE Winter Conference on Applications of Computer Vision, WACV 2021, 2021. doi: 10.1109/WACV48630.2021.00126.

[5] S. Sundaram, P. Kellnhofer, Y. Li, J. Y. Zhu, A. Torralba, and W. Matusik, "Learning the signatures of the human grasp using a scalable tactile glove," Nature, vol. 569, no. 7758, 2019, doi: 10.1038/s41586-019-1234-z.

[6] O. Ozioko and R. Dahiya, "Smart Tactile Gloves for Haptic Interaction, Communication, and Rehabilitation," Advanced Intelligent Systems, vol. 4, no. 2, 2022, doi: 10.1002/aisy.202100091.

[7] M. Altamirano Cabrera, J. Heredia, and D. Tsetserukou, "Tactile perception of objects by the user's palm for the development of multi-contact wearable tactile displays," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2020. doi: 10.1007/978-3-030-58147-3_6.

[8] P. K. Murali, C. Wang, D. Lee, R. Dahiya, and M. Kaboli, "Deep Active Cross-Modal Visuo-Tactile Transfer Learning for Robotic Object Recognition," IEEE Robot Autom Lett, vol. 7, no. 4, 2022, doi: 10.1109/LRA.2022.3191408.

[9] F. Ito and K. Takemura, "A model for estimating tactile sensation by machine learning based on vibration information

obtained while touching an object," Sensors, vol. 21, no. 23, 2021, doi: 10.3390/s21237772.

[10] X. Zhang et al., "Target classification method of tactile perception data with deep learning," Entropy, vol. 23, no. 11, 2021, doi: 10.3390/e23111537.

[11] S. Cai, K. Zhu, Y. Ban, and T. Narumi, "Visual-Tactile Cross-Modal Data Generation Using Residue-Fusion GAN with Feature-Matching and Perceptual Losses," IEEE Robot Autom Lett, vol. 6, no. 4, 2021, doi: 10.1109/LRA.2021.3095925.

[12] G. Rouhafzay, A. M. Cretu, and P. Payeur, "Transfer of learning from vision to touch: A hybrid deep convolutional neural network for visuo-tactile 3d object recognition," Sensors (Switzerland), vol. 21, no. 1, 2021, doi: 10.3390/s21010113.

[13] J. T. Lee, D. Bollegala, and S. Luo, "'Touching to see' and 'seeing to feel': Robotic cross-modal sensory data generation for visual-tactile perception," in Proceedings - IEEE International Conference on Robotics and Automation, 2019. doi: 10.1109/ICRA.2019.8793763.

[14] J. Lin, R. Calandra, and S. Levine, "Learning to identify object instances by touch: Tactile recognition via multimodal matching," in Proceedings - IEEE International Conference on Robotics and Automation, 2019. doi: 10.1109/ICRA.2019.8793885.

[15] X. Li, H. Liu, J. Zhou, and F. Sun, "Learning cross-modal visual-tactile representation using ensembled generative adversarial networks," Cognitive Computation and Systems, vol. 1, no. 2, 2019, doi: 10.1049/ccs.2018.0014.

[16] Y. Li, J. Y. Zhu, R. Tedrake, and A. Torralba, "Connecting touch and vision via cross-modal prediction," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2019. doi: 10.1109/CVPR.2019.01086.

[17] S. Pohtongkam and J. Srinonchat, "Tactile object recognition for humanoid robots using new designed piezoresistive tactile sensor and dcnn," Sensors, vol. 21, no. 18, 2021, doi: 10.3390/s21186024.

[18] P. Falco, S. Lu, A. Cirillo, C. Natale, S. Pirozzi, and D. Lee, "Cross-modal visuo-tactile object recognition using robotic active exploration," in Proceedings - IEEE International Conference on Robotics and Automation, 2017. doi: 10.1109/ICRA.2017.7989619.

[19] G. Izatt, G. Mirano, E. Adelson, and R. Tedrake, "Tracking objects with point clouds from vision and touch," in Proceedings - IEEE International Conference on Robotics and Automation, 2017. doi: 10.1109/ICRA.2017.7989460.

[20] M. Lambeta et al., "DIGIT: A Novel Design for a Low-Cost Compact High-Resolution Tactile Sensor with Application to In-Hand Manipulation," IEEE Robot Autom Lett, vol. 5, no. 3, 2020, doi: 10.1109/LRA.2020.2977257.

[21] S. Wang, M. Lambeta, P. W. Chou, and R. Calandra, "TACTO: A Fast, Flexible, and Open-Source Simulator for High-Resolution Vision-Based Tactile Sensors," IEEE Robot Autom Lett, vol. 7, no. 2, 2022, doi: 10.1109/LRA.2022.3146945.

[22] F. Rajeena P. P., A. S. U., M. A. Moustafa, and M. A. S. Ali, "Detecting Plant Disease in Corn Leaf Using EfficientNet Architecture—An Analytical Approach," Electronics (Basel), vol. 12, no. 8, p. 1938, Apr. 2023, doi: 10.3390/electronics12081938.

[23] P. K. Allen and K. S. Roberts, "Haptic object recognition using a multi-fingered dextrous hand," in Proceedings - IEEE International Conference on Robotics and Automation, 1989. doi: 10.1109/robot.1989.100011.

[24] S. Caselli, C. Magnanini, and F. Zanichelli, "On the robustness of haptic object recognition based on polyhedral shape representations," in IEEE International Conference on Intelligent Robots and Systems, 1995. doi: 10.1109/iros.1995.526160.

[25] Z. Pezzementi, C. Reyda, and G. D. Hager, "Object mapping, recognition, and localization from tactile geometry," in Proceedings - IEEE International Conference on Robotics and Automation, 2011. doi: 10.1109/ICRA.2011.5980363.

[26] M. Meier, M. Schöpfer, R. Haschke, and H. Ritter, "A probabilistic approach to tactile shape reconstruction," IEEE Transactions on Robotics, vol. 27, no. 3, 2011, doi: 10.1109/TRO.2011.2120830.

[27] A. Aggarwal, P. Kampmann, J. Lemburg, and F. Kirchner, "Haptic object recognition in underwater and deep-sea environments," J Field Robot, vol. 32, no. 1, 2015, doi: 10.1002/rob.21538.

[28] V. K. Nanayakkara, G. Cotugno, N. Vitzilaios, D. Venetsanos, T. Nanayakkara, and M. N. Sahinkaya, "The Role of Morphology of the Thumb in Anthropomorphic Grasping: A Review," Frontiers in Mechanical Engineering, vol. 3. 2017. doi: 10.3389/fmech.2017.00005.

[29] J. Bimbo, S. Luo, K. Althoefer, and H. Liu, "In-Hand Object Pose Estimation Using Covariance-Based Tactile To Geometry Matching," IEEE Robot Autom Lett, vol. 1, no. 1, 2016, doi: 10.1109/LRA.2016.2517244.

[30] J. M. Gandarias, A. J. Garcia-Cerezo, and J. M. Gomez-De-Gabriel, "CNN-Based Methods for Object Recognition with High-Resolution Tactile Sensors," IEEE Sens J, vol. 19, no. 16, 2019, doi: 10.1109/JSEN.2019.2912968.

[31] S. J. Pan and Q. Yang, "A survey on transfer learning," IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 10. 2010. doi: 10.1109/TKDE.2009.191.

[32] Y. Ganin et al., "Domain-adversarial training of neural networks," Journal of Machine Learning Research, vol. 17, 2016.

[33] G. Csurka, "A comprehensive survey on domain adaptation for visual applications," in Advances in Computer Vision and Pattern Recognition, 2017. doi: 10.1007/978-3-319-58347-1_1.

[34] P. Isola, J. Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, 2017. doi: 10.1109/CVPR.2017.632.

[35] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2010. doi: 10.1007/978-3-642-15561-1_16.

[36] Pekka Koskinen, Pieter van der Meer, Michael Steiner, Thomas Keller, Marco Bianchi. Automated Feedback Systems for Programming Assignments using Machine Learning. Kuwait Journal of Machine Learning, 2(2). Retrieved from http://kuwaitjournals.com/index.php/kjml/article/view/190

[37] Juhani Nieminen , Johan Bakker, Martin Mayer, Patrick Schmid, Andrea Ricci. Exploring Explainable AI in Educational Machine Learning Models. Kuwait Journal of Machine Learning, 2(2). Retrieved from http://kuwaitjournals.com/index.php/kjml/article/view/191

[38] Vadivu, N. S., Gupta, G., Naveed, Q. N., Rasheed, T., Singh, S. K., & Dhabliya, D. (2022). Correlation-based mutual information model for analysis of lung cancer CT image. BioMed Research International, 2022, 6451770. doi:10.1155/2022/6451770