Fundamentals of Neuroscience

Instructors: Prof. Reza Ebrahimpour - Prof. Sahar Kiani

Sharif University of Technology

Analysis of Hmax Model for Object Recognition Task and Evaluation of a Confidence Metric

---

Authors:

Mahdi Tabatabaei — Tabatabaeii.Mahdii@gmail.com
Heliya Shakeri — HeliyaaShakeri@gmail.com

Fall 2024

Preface

Notes on the project:

- Due date: 1403/11/15

- Psychophysics task due date: 1403/10/14

- The project must be done in groups of two individuals.

- Please submit your project report as a `.pdf` file. Include all outputs and final results in the report. The report must be in paper format (LaTeX or Word) and contain answers to all questions about the project.

- Please upload a one-minute recorded video of your psychophysics task, that one of the group members explains both the code and the task.

- Ensure that all codes are provided in a separate `.m/.py/.ipynb` file. If a code cannot be tested accurately upon submission, the reported results will be considered invalid, and no points will be awarded in such cases.

- You have the flexibility to utilize either `MATLAB` or `Python` for your project.

- Ensure that you save all files, including your report, codes, helper functions, and any additional outputs, if required, in a compressed file format such as `.zip` or `.rar`. This compressed file should then be uploaded to the Coursework CW submission platform.

- The project should be uploaded by one of the group members.

- Your file names must be in the following format:

  `Project_#StudentID_#StudentID.zip/.rar/.pdf/.m/.py/.ipynb/.mp4`

- The details of the grading system of this project is provided in Figure 9. Generally, the project is worth a total of 4 point, with an additional 1 point allocated for the bonus parts.

- In this project, it is essential to uphold the principles of academic integrity and refrain from any form of cheating or copying. Cheating undermines the learning process, diminishes personal growth, and compromises the trust placed in us as students/researchers/professionals. It is crucial to recognize that engaging in dishonest practices not only tarnishes our own reputation but also has serious consequences, both ethically and academically. We want to emphasize that if anyone is found to have cheated, their results will not be accepted in this project.

Contents

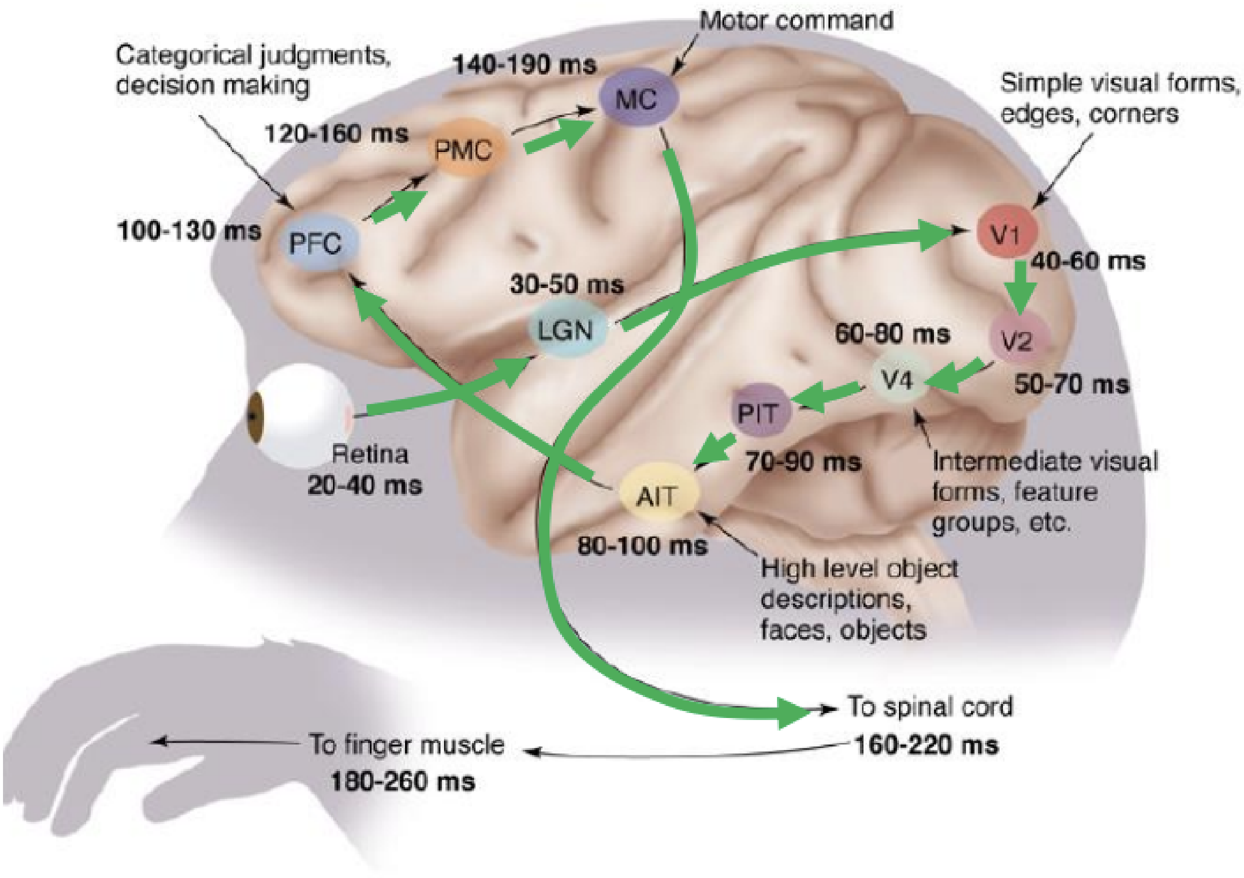

Figure 1: Ventral and Dorsal Visual Pathways

1 Introduction

Visual neuroscience focuses on understanding how the brain processes visual information, enabling us to perceive the world. This field explores the structure and function of the visual system, including the eye, retina, and brain regions responsible for vision. Visual input begins at the retina, where light is converted into electrical signals. These signals travel through the optic nerve to the lateral geniculate nucleus (LGN) and then to the primary visual cortex (V1), where basic features like edges, orientation, and motion are processed. Advanced processing occurs in higher cortical areas like V4 and the inferotemporal cortex, which help in recognizing complex patterns, colors, and objects. Visual neuroscience bridges the gap between biology and perception, explaining how neural pathways transform raw sensory data into meaningful visual experiences [10].

As you can see in figure 1 The **dorsal** and **ventral pathways** are two distinct visual processing streams in the brain, originating from the primary visual cortex (V1) and serving different roles in perception and action.

- The **dorsal pathway**, also known as the "Where" or "How" pathway, extends from the **V1** through the **parietal lobe** and is responsible for processing spatial information, motion, and guiding actions. It determines an object's location, movement, and how to interact with it, such as catching a ball by calculating its trajectory in real time. Damage to the dorsal stream can lead to deficits like difficulty in perceiving motion or spatial neglect [2].

- The **ventral pathway**, also known as the "What" pathway, projects from the **V1** to the **inferior temporal cortex** and specializes in object recognition and form perception. It identifies "what" an object is by analyzing features like color, shape, and texture. This pathway is essential for recognizing faces, objects, and visual details. Damage to the ventral stream can cause conditions like **visual agnosia**, where individuals cannot recognize objects despite intact vision [2].

Together, these pathways work in parallel to integrate *what* we see with *where* it is and *how* to act upon it, enabling seamless interaction with our environment.

1.1 Object Recognition

Object recognition is the ability of the visual system to identify and differentiate objects, a process fundamental to interacting with the environment. This begins with the perception of simple visual features such as edges, lines, and orientations in the primary visual cortex (V1). As visual information advances hierarchically, areas like V2 and V4 integrate these basic features into intermediate forms, such as shapes and contours. Eventually, the anterior and posterior inferior temporal (AIT and PIT) cortices process this information to identify whole objects, faces, or complex stimuli. The system achieves invariance, meaning it can recognize objects despite changes in size, lighting, angle, or position. Such robustness is achieved through hierarchical pooling of features across neural layers, where increasing complexity allows for accurate recognition under varying visual conditions [10].

Neuroscientific models of object recognition are inspired by both biological systems and computational approaches. For instance, feedforward models describe how visual signals move sequentially through neural layers, with each stage building upon the previous. Studies such as those by Riesenhuber and Poggio (1999) [6] demonstrate that neurons progressively encode more complex features while maintaining selectivity for specific objects. Additionally, the visual system relies on mechanisms such as context effects and priming, where prior knowledge or surrounding stimuli influence recognition speed and accuracy.

1.2 Decision-Making

Decision making is the process of evaluating alternatives and selecting an action. We tend to search for information related to the problem at hand, estimate the probabilities of different alternatives, and attach meanings and values to anticipated outcomes. Therefore, decisions are a choice among courses of actions. People who must make too many decisions too quickly have to trade off speed of decision making against accuracy of decision outcome. In addition, the time to make an accurate decision is related to the amount of uncertainty in the decision. Naturally, the more uncertainty we have, the longer it takes us to search for the information, estimate probabilities of different alternatives, and attach values to outcomes. [1].

1.3 Confidence

Confidence is the belief in the accuracy or correctness of a decision, action, or outcome. When we make choices, confidence reflects how sure we are that our decision is the right one. For example, if you answer a question and feel very sure it's correct, your confidence level is high. Confidence can be influenced by experience, the information available, and how often we've been correct in similar situations before. In the brain, confidence often involves areas like the prefrontal cortex, which processes reasoning and decision-making, and it helps guide actions by assessing how likely we are to succeed. [3].

1.4 Goal of the Project

This project is designed to enhance your understanding of the human brain's visual system. Through this project, you will learn to design a psychophysics task, collect behavioral data, and analyze the results. Additionally, you will become familiar with the HMAX model, a computational model inspired by the visual cortex. You will explore how this model extracts visual features and, after conducting analyses, compare its results with human behavioral data. Moreover, you will calculate confidence levels using this model.

2 Hmax Model

The **HMAX model** (Hierarchical Model and X) is a biologically inspired computational model of object recognition, designed to replicate the hierarchical feedforward processing observed in the primate visual cortex. It was initially developed by *Riesenhuber and Poggio* and has since been refined in numerous studies [7], [8].

2.1 Overview of the Model

The HMAX model mirrors the functionality of the ventral visual pathway, specifically the primary visual cortex (V1) to the inferotemporal cortex (IT). It uses a hierarchical approach where features are processed in alternating layers of *simple (S)* and *complex (C)* units. The alternating structure allows the model to achieve a tradeoff between *selectivity* (identifying specific features) and *invariance* (robustness to transformations like position, scale, and rotation).

- **S Layers:** Simple units perform feature extraction using a *Gaussian tuning operation*. They detect specific patterns such as edges, orientations, and contours.
- **C Layers:** Complex units pool information from S units via a *max operation*, enabling position and scale invariance.

2.2 How the Model Works

The HMAX model processes images through the following stages:

1. **S1 Layer (Input Stage):** The input image is analyzed by S1 units, which detect low-level features such as edges at specific orientations and scales, similar to V1 simple cells.
2. **C1 Layer (Complex Representation):** Outputs from S1 units are pooled to form position- and scale-invariant representations, resembling the behavior of V1 complex cells.
3. **S2 Layer (Intermediate Complexity):** At this stage, S2 units detect combinations of features such as corners and contours by pooling responses from C1 units.
4. **C2 Layer:** C2 units further pool S2 outputs across positions and scales, producing a dictionary of robust, invariant features.
5. **Higher Layers (S3 and Beyond):** The final stages (e.g., S3) integrate increasingly complex features, akin to visual processing in higher cortical areas like V4 and IT.

2.3 Biological and Computational Insights

- **Rapid Categorization:** It explains the brain's ability to quickly classify objects, even in cluttered or degraded scenes.
- **Robustness:** The model handles transformations such as changes in position, scale, and rotation effectively.
- **Feature Learning:** Using unsupervised learning, the model adapts to the statistics of natural images, tuning its feature detectors for better performance.

2.4 Code Availability

The MATLAB implementation of the HMAX model, along with related resources, is available for download. You can access the code at the following link: HMAX Model Code.

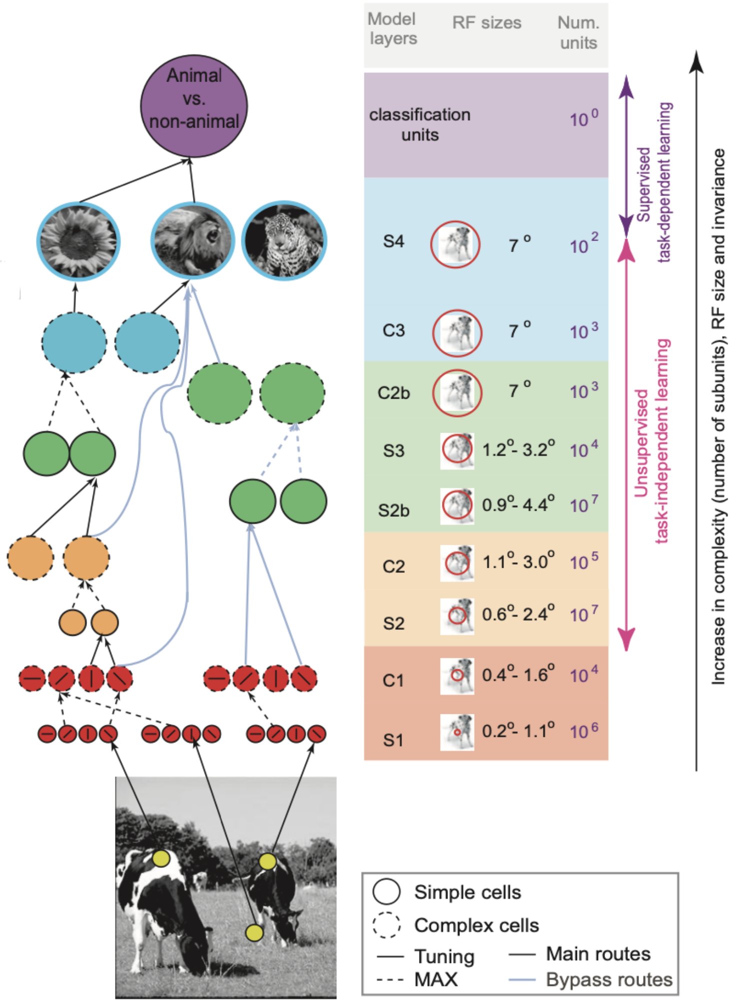

Figure 2: An illustration of the HMAX model structure, showing the hierarchical flow of information from simple to complex representations.

3 Dataset

3.1 Dataset Description

The dataset consists of grayscale 256 × 256 pixel images, divided into 600 animal stimuli and 600 non-animal stimuli. Animal stimuli are split into four categories: head, close-body, medium-body, and far. In the category of non-animal stimuli, there are 300 images of natural scenes and 300 images of artificial scenes, which are divided into four groups according to the mean distance from the camera: head (≤1 m), close-body (5–20 m), medium-body (50–100 m), and far-body (≥100 m).

The set of stimuli is divided into two separate training and test sets, with the same number of exemplars in each of the four mentioned groups, as presented in Figure 3 The dataset is divided into two subsets. Each subset contains 600 images with an equal number of images in each category. We used one of the subsets as a training set and the other one as the test set.

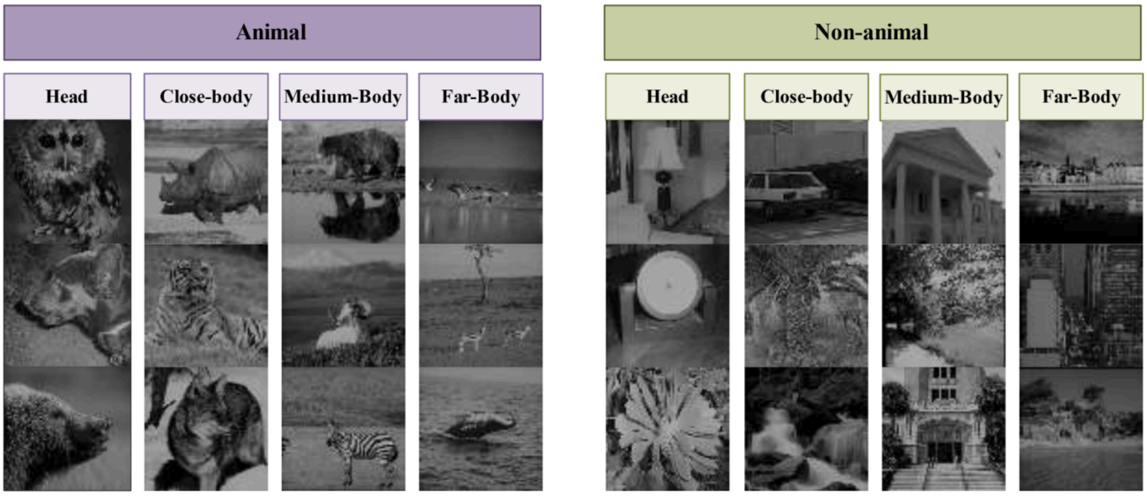

Figure 3: A Few Examples of The Dataset

3.2 Dataset Accessibility

You can access the dataset via this link.

4 Behavioral Task

In this section of, you will implement a MATLAB script using `Physchtoolbox` to create a task for the subjects to perform. Then, you will analyze the results.

- You can use this website to download and install `Physchtoolbox`.
- You can use this videos to learn how to implement a psychophysics task.
- You can use `Psychopy` in Python instead of `Physchtoolbox` for this part.
- Your implementation must be designed to calculate the accuracy, the mean response time, and the mean confidence for all stimuli and for specific categories (Head, Near-Body, Middle-Body, and Far-Body) after each block.
- **Note:** You must collect your own data from **at least two subjects**. For further analysis and to obtain significant results, you may also use data from other groups, provided you acknowledge the contributors by name.

4.1 Task Paradigm

This experiment consists of 1200 trials, divided into **10 blocks**, with each block containing **120 trials**. Subjects should be allowed to rest between blocks. In the first 5 blocks, training images will be used, while the remaining blocks will use test images. Each trial consists of a sequence of events in which either an animal or a non-animal image is **randomly** presented. Figure 4 shows the task paradigm and the sequence for each trial is as follows:

1. Fixation Cross: A plus sign (+) will appear in the center of the screen for 500 ms.
2. Stimulus Presentation: After the fixation cross disappears, a stimulus (either an animal or non-animal image) will be shown for 20 ms.
3. Gray Screen (ISI): The screen will then display a gray color for 30 ms.
4. Mask: A masked version of the image will be shown for 80 ms.
   Note: Masked version of images can be created by scrambling the pixels of the original stimulus.
5. Decision Screen: Subjects are then asked to make a decision regarding the presented image was an "Animal" or a "Non-Animal." The screen displays two color bars as columns: if their choice is "Animal," they should select the right column, and for "Non-Animal," they should select the left column. The height of the bar indicates their confidence level, with a higher green portion representing greater confidence and a higher red portion representing lower confidence. Confidence height should be between 0 and 1.

*Bonus: In this part, we aim to evaluate the subjects' performance in challenging visual scenarios. To achieve this, we will implement two tasks:

1. Add Gaussian noise to the test images and collect data for the last two blocks.
2. Rotate the test images by a random angle and collect data for the last two blocks.

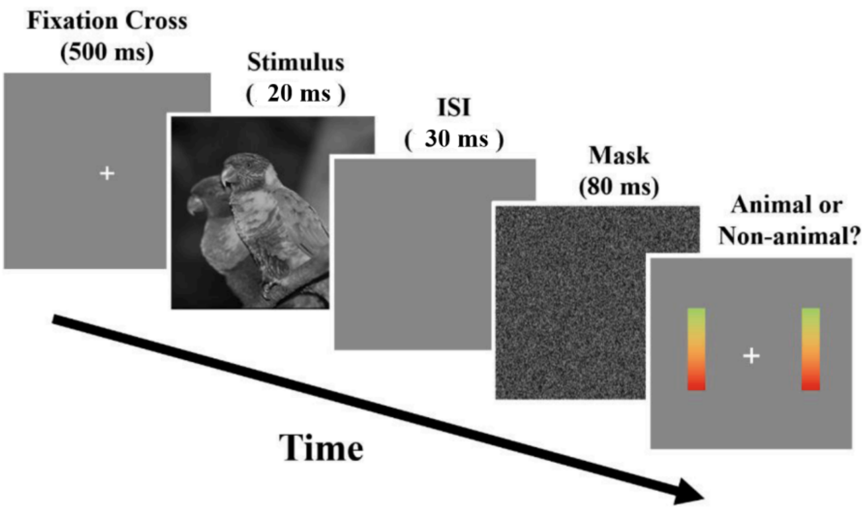

Figure 4: Behavioral Task Paradigm

4.2 Subject Results

After collecting the data,

1. Report all subject's performances on test images in a format similar to Table 1. Each subject must have three separate tables: one for accuracy, one for response time, and one for confidence.
2. Plot accuracy, response time, and confidence for each category for all subjects on the test images, similar to Figure 5.
3. Complete parts 1 and 2 for the analysis of the mean performance of the subjects.

	All Images	Head	Near-Body	Middle-Body	Far-Body
Original images					
Rotation					
Noise					

Table 1: Mean Accuracy of Subjects

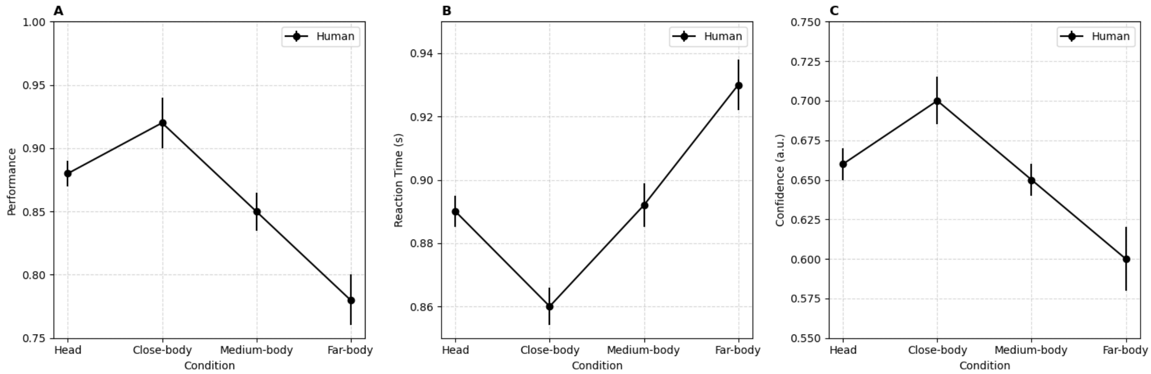

Figure 5: Sample Figure for Each Condition' Performance

4.3 Questions

- Why the stimulus presentation period is very short?
- What is role of ISI in the paradigm?
- What is role of Mask in the paradigm?
- What were your challenges during implementing this task and collecting data from subjects? How did you overcome with them?
- *Bonus: What was the effect of challenging scenarios (Noise/Rotation) on accuracy, response time, and confidence?

5 Model Training and Evaluations

5.1 Training the Model

To train the HMAX model, the provided MATLAB code demonstrates how to extract C2 features. Follow the steps below to set up and run the code:

1. **Download the Code:** First, download the MATLAB implementation of the HMAX model from the following link: HMAX Model Code. Ensure that the downloaded files include the `demoRelease.m` script. You can read the code explanation here.

2. **Add Paths to Required Directories:** Open the `demoRelease.m` file and modify the paths for the image directories to match your system setup. Specifically, you need to update the following lines:

```
train_set.pos    = 'Dataset/Train/Train_Animals';
train_set.neg    = 'Dataset/Train/Train_Non-Animals';
test_set.pos     = 'Dataset/Test/Test_Animals';
test_set.neg     = 'Dataset/Test/Test_Non-Animals';
```

   Replace the above paths with the correct locations of your image datasets.

3. **Run the Script:** Execute the script `demoRelease.m` in MATLAB. The script performs the following tasks:

   - Loads training and testing images from the specified directories.
   - Extracts C1 prototypes (low-level features) and uses them to compute C2 features.
   - Configures the settings for Gabor filters, scales, and receptive field sizes.
   - Outputs C2 features which can then be used for training and testing classifiers.

5.2 Classification

In this part, you will apply classifiers to the C2 features obtained from the HMAX model. Classifiers are machine learning algorithms used to assign labels to data based on its features. We will focus on two widely used classifiers: *Support Vector Machines (SVM)* and *Multi-Layer Perceptron (MLP)*.

5.2.1 Support Vector Machine (SVM)

The **Support Vector Machine** [4] is a supervised learning algorithm that seeks to find a hyperplane in a high-dimensional feature space that best separates two classes. For linearly separable data, the goal is to maximize the margin between the hyperplane and the closest data points (support vectors).

The decision function of an SVM can be formulated as follows:

$$f(\mathbf{x}) = \mathrm{sign}(\mathbf{w}^T\mathbf{x} + b), \tag{1}$$

where $\mathbf{w}$ is the weight vector, $\mathbf{x}$ is the input feature vector, and b is the bias term. The optimization objective is:

$$\min_{\mathbf{w},b} \quad \frac{1}{2}||\mathbf{w}||^2 \quad \text{subject to} \quad y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, \forall i, \tag{2}$$

where y_i is the class label ($+1$ or -1) for the data point $\mathbf{x}_i$.

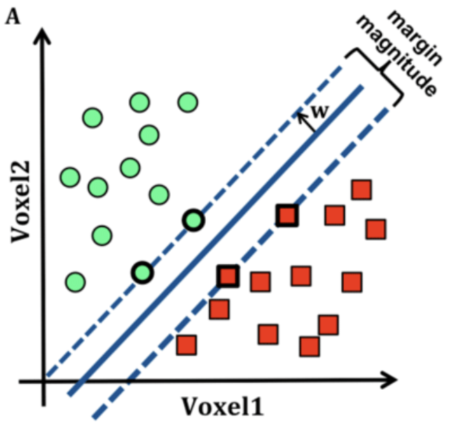

Figure 6: Support Vector Machine (SVM): A hyperplane separating two classes with the margin maximized.

5.2.2 Multi-Layer Perceptron (MLP)

The **Multi-Layer Perceptron** [9] is a type of neural network consisting of an input layer, one or more hidden layers, and an output layer. Each layer consists of neurons connected by weighted edges, with non-linear activation functions applied at each node.

The forward pass of an MLP is given by:

$$\mathbf{h} = \sigma(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1), \quad \mathbf{y} = \sigma(\mathbf{W}_2\mathbf{h} + \mathbf{b}_2), \tag{3}$$

where $\mathbf{x}$ is the input feature vector, $\mathbf{W}_1$ and $\mathbf{W}_2$ are the weight matrices, $\mathbf{b}_1$ and $\mathbf{b}_2$ are the bias terms, σ is the activation function, and $\mathbf{h}$ and $\mathbf{y}$ represent the hidden and output layer activations, respectively.

5.2.3 Task

Use the SVM and MLP classifiers to train, validate, and test on the C2 features extracted in the previous section. Compare the classification performance of both methods using the results from the test set.

5.3 Evaluation of the Performance

Evaluate the performance of the classifiers using the following metrics for the test sets:

- **Accuracy:** The proportion of correctly classified instances.
- **Confusie and AUC:** Plot the Receiver Operating Characteristic (ROC) curve and compute the Area Under the Curve (AUC).

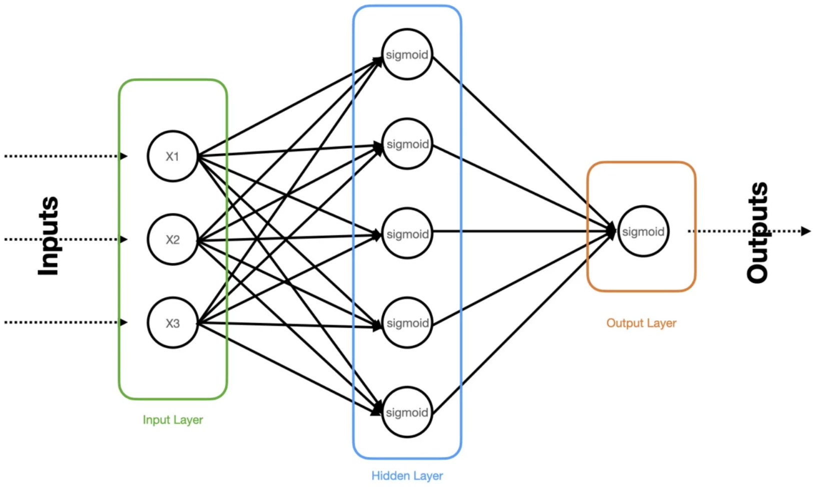

Figure 7: Structure of a Multi-Layer Perceptron (MLP) with input, hidden, and output layers.

5.4 Task

1. Compare the results of the SVM and MLP classifiers in the following table:

Table 2: Comparison of Classifiers on Test Set

Metric	SVM	MLP
Accuracy		
AUC		

2. Which classifier performs better on the given data?
3. Calculate the accuracy for all categories (Head, Near-Body, Middle-Body, Far-Body) and report them.
4. Plot the accuracy for each category similar to Figure 5.
5. How does changing the hyperparameters (e.g., kernel type for SVM, hidden layers for MLP) affect the results?
6. What does the ROC curve show?
7. What are the advantages and disadvantages of SVM and MLP classifiers?

*Bonus: In this part, you will evaluate the robustness of the HMAX model by testing it on challenging visual scenarios:

1. Add Gaussian noise to the test images and use them as input to the HMAX model.
2. Rotate the test images by a random angle and use them as input to the HMAX model.

Table 3: Comparison of Classifiers on Modified Test Sets

Metric	SVM (Noisy)	SVM (Rotated)	MLP (Noisy)	MLP (Rotated)
Accuracy				
AUC				

Repeat the classification tasks using the SVM and MLP classifiers on these modified test images. Evaluate the performance metrics (accuracy, confusion matrix, and ROC curve) and compare the results to those obtained from normal test images. Use the following table for the comparison:

3. How do the results from the modified test sets compare to the results from the normal test set? Provide an explanation for the observed differences.
4. Which classifier is more robust to noise and rotation? Why?
5. How might the HMAX model be improved to handle noisy or rotated images better?

5.5 Effect of Dimension Reduction

Dimension reduction is a technique used to reduce the number of features in a dataset while retaining as much of the original variance as possible. By reducing dimensions, we aim to simplify the dataset, improve computational efficiency, and potentially enhance model performance by eliminating redundant or irrelevant features. One commonly used method for dimension reduction is **Principal Component Analysis (PCA)**.

5.5.1 Principal Component Analysis (PCA)

PCA [5] is a statistical technique that transforms a dataset into a new coordinate system such that the greatest variances in the data are represented along the first principal components. Mathematically, PCA works as follows:

1. **Standardize the Dataset:** Subtract the mean from each feature and scale to unit variance to ensure that all features contribute equally.
2. **Compute the Covariance Matrix:** Calculate the covariance matrix to understand the relationships between different features.

$$\mathbf{C} = \frac{1}{n-1}\sum_{i=1}^{n}(\mathbf{x}_i - \mu)(\mathbf{x}_i - \mu)^T, \tag{4}$$

   where $\mathbf{x}_i$ is the i-th data point, μ is the mean vector, and n is the total number of data points.
3. **Compute Eigenvalues and Eigenvectors:** Solve the eigenvalue equation:

$$\mathbf{C}\mathbf{v} = \lambda\mathbf{v}, \tag{5}$$

   where λ represents the eigenvalues (amount of variance captured) and $\mathbf{v}$ represents the eigenvectors (principal components).

4. **Select Principal Components:** Order the eigenvalues in descending order and select the top k components that capture the most variance.

5. **Transform the Data:** Project the data onto the selected components to create a reduced dataset:
$$\mathbf{Z} = \mathbf{X}\mathbf{W}, \tag{6}$$
where $\mathbf{W}$ is a matrix of the selected eigenvectors.

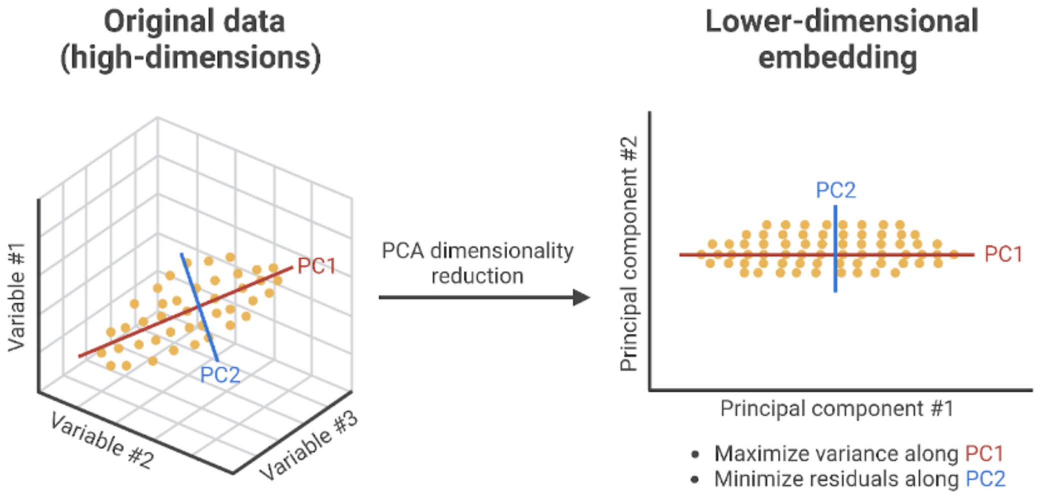

Figure 8: PCA Dimensionality Reduction: High-dimensional data is projected onto lower-dimensional principal components.

5.5.2 Task and Analysis

1. Apply PCA to the C2 features, selecting principal components to retain at least 95% of the variance.

2. Train and test the SVM and MLP classifiers using the reduced-dimension dataset.

3. Evaluate the performance of the classifiers in terms of accuracy, confusion matrix, and AUC.

4. Compare the results with those obtained from the original full-dimension dataset in a table:

Table 4: Comparison of Classifiers Before and After PCA

Metric	SVM (Original)	SVM (PCA)	MLP (Original)	MLP (PCA)
Accuracy				
AUC				

5.5.3 Questions

1. How does PCA affect the performance of the SVM and MLP classifiers? Does it improve or degrade performance?
2. How does the computational efficiency change after applying PCA?
3. What are the advantages and disadvantages of using PCA for dimension reduction in this context?

6 Exploring Confidence

In this part, we aim to define and calculate a confidence metric based on the HMAX model's output. Confidence will be evaluated by analyzing the output of the classifier for different inputs.

6.1 Task

In the previous section, you implemented SVM and MLP classifiers, where the outputs were likely binary. In this section, you will extend the classifiers to provide probability outputs for each class and calculate a confidence score.

1. **Implement a classifier that provides probability outputs for each class.** For example, the output should indicate probabilities such as 0.7 for *animal* and 0.3 for *non-animal*.

2. **Calculate the absolute difference of the outputs for each class.** For a given input, if the probabilities are 0.7 (animal) and 0.3 (non-animal), the confidence score will be:
$$\text{Confidence} = |P_{\text{animal}} - P_{\text{non-animal}}|. \tag{7}$$

3. **Calculate the mean confidence of the model for the entire dataset.** Aggregate the confidence scores obtained for all test samples and compute their mean:
$$\text{Mean Confidence} = \frac{1}{N}\sum_{i=1}^{N} \text{Confidence}_i, \tag{8}$$
where N is the total number of test samples.

4. Calculate the confidence metric for all categories (Head, Near-Body, Middle-Body, Far-Body).

5. Plot the confidence level for each category similar to Figure 5.

6. Use PCA to reduce input vectors' dimension and then calculate the confidence and plot it for all categories again.

7. *Bonus Use the noisy and rotated images to calculate the confidence of the model for these images. Plot the confidence for all categories as well. (PCA is not needed for this part.)

6.2 Questions

1. Compare the confidence scores for the SVM and MLP classifiers and discuss the results. Which classifier provides higher confidence?

2. How does confidence relate to classification accuracy?

3. What is the effect of PCA on confidence level?

4. What is the effect of noise and rotation on confidence level?

References

[1] Joshua I Gold and Michael N Shadlen. The neural basis of decision making. *Annual review of neuroscience*, 30:535—574, 2007.

[2] Melvyn A. Goodale and A.David Milner. Separate visual pathways for perception and action. *Trends in Neurosciences*, 15(1):20–25, 1992.

[3] Piercesare Grimaldi, Hakwan Lau, and Michele Basso. There are things that we know that we know, and there are things that we do not know we do not know: Confidence in decision-making. *Neuroscience and biobehavioral reviews*, 55, 04 2015.

[4] M.A. Hearst, S.T. Dumais, E. Osuna, J. Platt, and B. Scholkopf. Support vector machines. *IEEE Intelligent Systems and their Applications*, 13(4):18–28, 1998.

[5] Andrzej Maćkiewicz and Waldemar Ratajczak. Principal components analysis (pca). *Computers Geosciences*, 19(3):303–342, 1993.

[6] Maximilian Riesenhuber and Tomaso Poggio. Riesenhuber, m. & poggio, t. hierarchical models of object recognition in cortex. nat. neurosci. 2, 10191025. *Nature neuroscience*, 2:1019–25, 12 1999.

[7] Thomas Serre, Aude Oliva, and Tomaso Poggio. A feedforward architecture accounts for rapid categorization. *Proceedings of the National Academy of Sciences*, 104(15):6424–6429, 2007.

[8] Thomas Serre, Lior Wolf, Stanley Bileschi, Maximilian Riesenhuber, and Tomaso Poggio. Robust object recognition with cortex-like mechanisms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(3):411–426, 2007.

[9] Gurpreet Singh and Manoj Sachan. Multi-layer perceptron (mlp) neural network technique for offline handwritten gurmukhi character recognition. In *2014 IEEE International Conference on Computational Intelligence and Computing Research*, pages 1–5, 2014.

[10] Simon J. Thorpe and Michèle Fabre-Thorpe. Seeking categories in the brain. *Science*, 291:260 – 263, 2001.

Section	Task	Score
Psychophysics Task	Task Paradigm (2 Subjects)	60
	Subjects' Results (Table 1)	30
	Subjects' Mean Results	10
	Condition Based Results (Fig. 5)	40
	Questions	30
	Video	30
	Noisy Images	25
	Rotated Images	25
Hmax Model	SVM Classifier Evaluation	20
	MLP Classifier Evaluation	20
	Category Based Output	40
	Performance on Noisy Images	15
	Performance on Rotated Images	15
	PCA + SVM Evaluation	15
	PCA + MLP Evaluation	15
	Confidence with SVM	15
	Confidence with MLP	15
	Category Based Output	20
	Confidence on Noisy Images	10
	Confidence on Rotated Images	10
	Questions	40
Project	**Sum**	**500**

Figure 9: Grading System