

Fundamentals of Neuroscience Homework Report

Ehsan Merrikhi
Sharif University of Technology
ehsan.merrikhi@gmail.com

AmirAli Rostami
Sharif University of Technology
amiralirostami99@gmail.com

Abstract

This project explores the intersection of visual neuroscience and computational modeling by implementing a behavioral psychophysics task and analyzing its results. The study aims to investigate object recognition processes through human subjects and computational models inspired by the visual cortex. A MATLAB-based experimental task was designed using Psychtoolbox, featuring trials with varying stimuli to assess accuracy, response time, and confidence across specific categories. Additional tasks introduced challenging scenarios with noise and rotation to evaluate performance robustness. By comparing human data to computational outputs, this project provides insights into visual recognition, decision-making, and confidence metrics, emphasizing the synergy between neuroscience and computational methods.

1 Introduction

This report details the findings and implementation for the Fundamentals of Neuroscience project. The project involves studying object recognition models and implementing a behavioral psychophysics task.

2 Behavioral Task

In this section, we implement a MATLAB script using Psychtoolbox to create a task for the subjects to perform. Then, we will analyze the results.

2.1 Task Paradigm

2.1.1 Fixation Cross (500 ms)

- Participants are instructed to focus on a fixation cross at the center of the screen.
- This ensures that their attention is directed at the screen before the stimulus is presented.

2.1.2 Stimulus Presentation (20 ms)

- A black-and-white image is briefly flashed on the screen.
- The image may contain an animal or a non-animal object.
- In test phase some subjects see noise injected images for the last two blocks and other subjects see rotated images for the last two blocks.

2.1.3 3. Inter-Stimulus Interval (ISI) (30 ms)

- A blank gray screen is shown to minimize retinal persistence and prepare for the masking phase.
- This helps in ensuring that the processing of the stimulus occurs before masking.

2.1.4 4. Masking (80 ms)

- A noise mask (random pixel noise) is presented.
- This disrupts any lingering visual information from the stimulus to prevent afterimage effects.

2.1.5 5. Decision Screen (Until Response)

- A response screen appears with a fixation cross and two response indicators.
- Participants must quickly decide whether the image contained an animal or not.

- Responses are recorded via a button press or keypress.

2.1.6 Data Collection and Analysis

- Reaction Time (RT): Time taken to respond after the decision screen appears.
- Accuracy (% correct responses): Number of correct responses divided by total trials.
- Error Rate: Incorrect categorizations of animal/non-animal images.

2.2 Subject Results

	All Images	Head	Near-Body	Middle-Body	Far-Body
Original images	92.30%	96.73%	91.01%	93.97%	87.50%
Rotation	76.91%	81.96%	85.71%	81.08%	58.92%
Noise	94.66%	94.82%	98.36%	91.04%	94.44%

Table 1: Mean Accuracy of Subjects

	All Images	Head	Near-Body	Middle-Body	Far-Body
Original images	2.11	2.07	2.01	2.17	2.17
Rotation	0.78	0.71	0.73	0.83	0.87
Noise	1.92	1.93	1.91	1.90	1.94

Table 2: Mean Response Time of Subjects

	All Images	Head	Near-Body	Middle-Body	Far-Body
Original images	79.50%	81.30%	83.31%	79.39%	74.32%
Rotation	50.00%	50.00%	50.00%	50.00%	50.00%
Noise	78.97%	82.50%	79.42%	81.56%	71.48%

Table 3: Mean Confidence of Subjects

2.3 Questions

- Why is the stimulus presentation period very short?

The stimulus presentation period is very short to mimic rapid visual categorization processes, ensuring subjects rely on immediate perception rather than prolonged analysis. This helps simulate real-world rapid decision-making scenarios.

- What is the role of ISI in the paradigm?

The ISI (Inter-Stimulus Interval) serves to separate stimuli presentations, preventing overlapping visual or cognitive processes and allowing the brain to reset between trials.

- What is the role of the mask in the paradigm?

The mask disrupts the afterimage effect and prevents continued processing of the stimulus after its presentation, ensuring the decision is based solely on the brief exposure.

- What were your challenges during implementing this task and collecting data from subjects? How did you overcome them?

The challenges faced during data collection included establishing a controlled environment for consistent stimulus presentation, ensuring precise timing of ISI and mask durations, and minimizing external distractions that could affect participants' responses. Additionally, technical issues such as synchronization of response logging and stimulus display accuracy posed difficulties.

To overcome these challenges, we used precise timing functions in the experimental software to ensure accurate stimulus durations. Pilot testing was conducted to fine-tune timing parameters and verify synchronization. We also provided clear instructions to participants and minimized environmental distractions to ensure reliable data collection. Finally, data preprocessing techniques were implemented to filter out any noisy or incomplete responses.

- Bonus: What was the effect of challenging scenarios (Noise/Rotation) on accuracy, response time, and confidence?

It is assumed that noise and rotation decrease accuracy and confidence while increasing response times. The additional visual complexity challenges subjects' ability to identify stimuli, highlighting the robustness and limitations of human visual processing.

3 Model Training and Evaluations

3.1 Training the Model

In this section we train the HMAX model and extract C2 features

1. Download the Code

2. Add Paths to Required Directories

3. Run the Script

The output of the script would before
['X_train', 'Y_train', 'X_test', 'Y_test']

3.2 Classification

In this part, we will apply classifiers to the C2 features obtained from the HMAX model.

3.2.1 Support Vector Machine (SVM)

The Support Vector Machine is a supervised learning algorithm that seeks to find a hyperplane in a high-dimensional feature space that best separates two classes.

3.2.2 Multi-Layer Perceptron (MLP)

The Multi-Layer Perceptron 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.

3.2.3 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.

We trained our classifiers on the training data and evaluated their performance using AUC, accuracy, and the confusion matrix. While the SVM achieved higher accuracy and produced a better confusion matrix, the MLP outperformed the SVM in terms of AUC.

3.3 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)

3.4 Task

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

Metric	SVM	MLP
Accuracy	73.66%	72.00%
AUC	0.8034	0.8241

Table 4: Comparison of Classifiers on Test Set

2. Which classifier performs better on the given data?

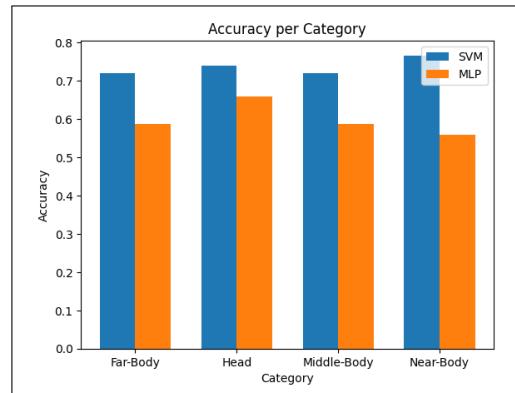
While the SVM achieved higher accuracy and produced a better confusion matrix, the MLP outperformed the SVM in terms of AUC.

3. Calculate the accuracy for all categories (Head, Near-Body, Middle-Body, Far-Body) and report them.

Accuracy	Head	Near-Body	Middle-Body	Far-Body
SVM (RBF Kernel)	74%	76%	72%	72%
MLP	66%	56%	58%	58%

Table 5: Comparison of Classifiers on Modified Test Sets

4. Plot the accuracy for each category.



5. How does changing the hyperparameters (e.g., kernel type for SVM, hidden layers for MLP) affect the results?

Changing hyperparameters significantly impacts model performance:

- SVM (Kernel Type): Affects decision boundary flexibility—linear kernels work well for simple data, while RBF captures complex patterns.
- MLP (Hidden Layers): More layers improve learning capacity but may lead to overfitting and higher computational cost.

Tuning these parameters balances accuracy, generalization, and efficiency.

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

This code is added to the HMAX model to inject gaussian noise.

```
% ADD GAUSSIAN NOISE TO TEST IMAGES
for i = 3:4
% Apply noise only to test images
for j = 1:length(cI{i})
    cI{i}{j} = imnoise(cI{i}{j}, ...
        'gaussian', 0, 0.01);
end
end
```

This code is added to the HMAX model to inject gaussian noise.

Listing 1: Rotate Test Images

```
% ROTATE TEST IMAGES
angle = 15;
for i = 3:4
    for j = 1:length(cI{i})
        cI{i}{j} = imrotate(cI{i}{j},
            angle, 'bilinear', 'crop');
    end
end
```

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.

Using noisy and rotated images on HMAX gives us the results shown in the table below

Metric	SVM (Noisy)	SVM (Rotated)	MLP (Noisy)	MLP (Rotated)
Accuracy	64.00%	70.33%	58.16%	59.15%
AUC	0.6879	0.7718	0.6793	0.7991

Table 6: Comparison of Classifiers on Modified Test Sets

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

The results on the modified test sets are significantly weaker than those on the normal tests in both accuracy and AUC. This performance degradation can be attributed to the HMAX model's sensitivity to noise and rotations. Additionally, this finding aligns with our understanding of human visual processing; our brains can distinguish normal images more easily than noisy or rotated ones because normal images are inherently more interpretable.

4. Which classifier is more robust to noise and rotation? Why?

SVM demonstrates greater robustness to noise and rotation. While both SVM and MLP experience a significant drop in accuracy, SVM maintains a considerably higher accuracy than MLP. In terms of AUC, the difference between the two models is minimal. Therefore, we conclude that SVM is more resilient to noise and rotation.

5. How might the HMAX model be improved to handle noisy or rotated images better?

To improve HMAX for noisy or rotated images:
<ul style="list-style-type: none"> • Data Augmentation: Train with noise and rotated images for better generalization. • Feature Enhancement: Optimize Gabor filters and use multi-scale processing. • Preprocessing: Apply denoising filters and affine normalization. • Hybrid Models: Combine HMAX with CNNs for improved robustness. • Adaptive C2 Layer: Modify pooling and introduce a voting mechanism. • Learning-Based Adaptations: Fine-tune on noisy data and integrate attention mechanisms.

3.5 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).

3.5.1 Principal Component Analysis (PCA)
PCA 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.

3.5.2 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:

Metric	SVM (Original)	SVM (PCA)	MLP (Original)	MLP (PCA)
Accuracy	73.66%	73.66%	72.00%	73.50%
AUC	0.8034	0.8093	0.8241	0.8099

Table 7: Comparison of Classifiers Before and After PCA

3.5.3 Questions

1. How does PCA affect the performance of the SVM and MLP classifiers? Does it improve or degrade performance?

Its impact on performance depends on the dataset—sometimes it improves generalization by removing noise, but it can also degrade performance if important features are lost. In this case PCA has not changed the performance of HMAX model.

2. How does the computational efficiency change after applying PCA?

PCA reduces computational cost by eliminating redundant features, which can help both SVM and MLP train faster.

3. What are the advantages and disadvantages of using PCA for dimension reduction in this context?

Advantages:

- Reduces computational cost and training time.
- Removes redundant and less important features.
- Can improve model generalization by reducing noise.

Disadvantages:

- Risk of losing important information.
- May not always improve classification performance.
- Harder to interpret transformed features.

4 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.

4.1 Task

In the previous section, you implemented SVM and MLP classifiers, where the outputs were likely binary.

1. Implement a classifier that provides probability outputs for each class.

In the previous section, we implemented binary classifiers. In this section, we will modify the classifiers to provide probability outputs for each class. Specifically, the output for a given sample should be a probability distribution across the classes, such as 0.7 for class "animal" and 0.3 for class "non-animal."

- We use classifiers such as SVM with probability outputs (using `probability=True`) and MLP to obtain these probabilities.
- These probability values will indicate how likely each class is for a given sample.

2. Calculate the absolute difference of the outputs for each class.

To calculate the confidence score, we use the absolute difference between the probabilities of the classes. For a given sample, the confidence score can be computed using the following formula:

$$\text{Confidence} = |P_{\text{class1}} - P_{\text{class2}}|$$

For example, if the probabilities are 0.7 for class "animal" and 0.3 for class "non-animal", the confidence score will be:

$$\text{Confidence} = |0.7 - 0.3| = 0.4$$

3. Calculate the mean confidence of the model for the entire dataset.

Next, we calculate the mean confidence of the model for the entire dataset. The mean confidence is obtained by averaging the confidence scores for all test samples:

$$\text{Mean Confidence} = \frac{1}{N} \sum_{i=1}^N \text{Confidence}_i$$

where N is the total number of test samples. The mean confidence values for the models are as follows:

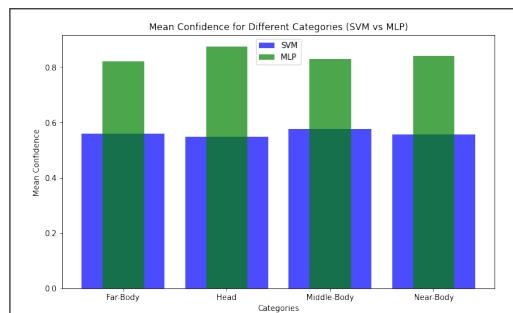
- Mean Confidence for SVM: 0.5709
- Mean Confidence for MLP: 0.8382
- Mean Confidence for SVM (PCA): 0.6211
- Mean Confidence for MLP (PCA): 0.8256

4. Calculate the confidence metric for all categories.

For different categories, such as "Head", "Near-Body", "Middle-Body", and "Far-Body", we will calculate the confidence score separately for each category. We will aggregate the confidence scores for each category and compute the mean confidence for that category.

- First, we filter the samples belonging to each category based on their indices.
- Then, we calculate the mean confidence for each category.

5. Plot the confidence level for each category.



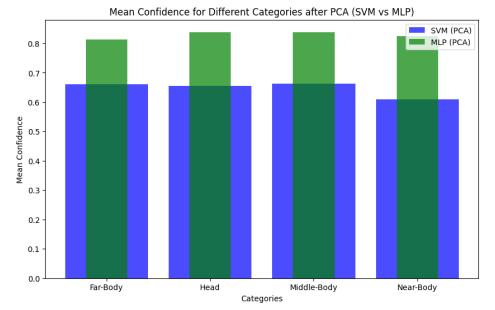
6. Use PCA to reduce input vectors' dimension and then calculate the confidence and plot it for all categories again.

Finally, we will apply Principal Component Analysis (PCA) to reduce the dimensionality of the input vectors and recalculate the confidence for each category. PCA will help retain 95% of the variance in the data.

- We apply PCA to both the training and testing datasets.
- After applying PCA, we recalculate the confidence scores for the SVM and MLP classifiers.
- We then plot the confidence scores for all categories again.

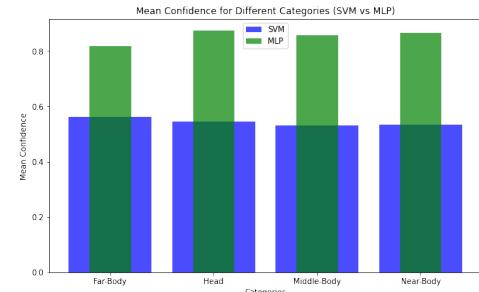
Mean Confidence for SVM (PCA):
0.6470

Mean Confidence for MLP (PCA):
0.8282

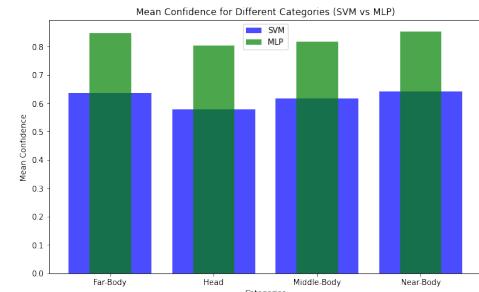


7. *Bonus Use the noisy and rotated images to calculate the confidence of the model for these images. Plot the confidence for all categories as well. (PCA is not needed for this part.)

Rotated data mean confidence results and plot for different categories:
Mean Confidence for SVM: 0.5431
Mean Confidence for MLP: 0.8540



Noisy data mean confidence results and plot for different categories:
Mean Confidence for SVM: 0.6183
Mean Confidence for MLP: 0.8290



4.2 Questions

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

MLP shows higher confidence scores than SVM because:

- Non-Linearity Handling: MLP, with its multiple layers and activation functions, captures complex patterns better, leading to stronger decision boundaries.
- Probability Estimates: MLP outputs softmax probabilities, making its confidence scores more distinct compared to SVM's margin-based decision.
- Feature Learning: MLP learns hierarchical feature representations, enhancing certainty in classification.

2. How does confidence relate to classification accuracy?

Confidence reflects how certain a model is in its predictions, but high confidence doesn't always mean high accuracy. A well-calibrated model balances confidence with accuracy, while overconfident models may misclassify with high certainty.

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

PCA can affect confidence levels in classification by reducing feature dimensionality:

- Increased Confidence: If PCA removes noise and redundant features, models may make clearer distinctions, leading to higher confidence.
- Decreased Confidence: If important features are lost, the model may struggle to differentiate classes, reducing confidence in predictions.

In this case PCA caused a growth in our confidence level.

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

Noise and rotation generally reduce confidence levels because they distort key features used for classification.

- Noise: Adds random variations, making it harder for the model to recognize patterns, leading to lower confidence.
- Rotation: Alters feature alignment, especially affecting models sensitive to spatial structure, reducing certainty in predictions.

If a model is robust, confidence may drop less, but significant distortions typically lead to lower confidence and accuracy.