each object, with a visibility level ranging from g=0.0 (no information about the source image) to g=0.3 (where subjects could easily identify the object from the source image) [4].

B. *Brain waves*

The relevant information extraction from the raw signals is a critical step in EEG pattern classification due to its direct influence on classification performance [2]. Most of the EEG waves range from 0.5-500Hz, however gamma bands have been associated with visual stimulation and in movement or motor tasks and were chosen to compute our matrix. The gamma-band activity comprises an EEG frequency range, from 30 to 100 Hz and is distributed widely throughout the cerebral structure.

The Delta waves frequency is up to 4Hz. It is the highest in amplitude and slowest in wave representing the grey matter of the brain. It is dominant in infants up to one year and adults in deep sleep.

The Theta waves with a frequency range from 4Hz to 8Hz are seen in young children and in adults and emerge with closing of the eyes and with relaxation.

The Alpha waves with frequency range from 8Hz to 14Hz are commonly seen in adults. They appear with closing eyes and disappear normally with opening eyes. Their appearance has also been linked to lack of stress. [6]

The beta activity has a frequency range between 14Hz and 30Hz and is dominant in subjects who are alert or anxious.

Gamma-band activity participates in various cerebral functions, such as perception, attention, memory, consciousness, motor control and it is possible to differentiate between low gamma-band oscillations (30-60Hz) and high gamma-band oscillations (>60Hz) [7].

III. APPROACH

In this section are presented the methods we used for feature extraction: segmentation of the EEG signal and then computing the features using sliding window and FFT.

In signal processing, time-frequency analysis is used to study a signal in both the time and frequency domains simultaneously using various time-frequency representations. In our case, we used the fast Fourier transform (FFT) with a sliding window of length 2048 samples and a step of 20 samples (see III.B) to compute the time-frequency power matrix which has 210 rows and 128 columns. The number of rows is computed using the formula

$$Objects\ number * Stimuli\ number = Rows\ number$$

and the number of columns is equal with the number of electrodes.

For classification, we used three supervised machine learning algorithms: Support Vector Machine (SVM), K-Nearest Neighbors (KNN) and Artificial Neural Networks (ANNs). The three possible outcomes of the recognition task represent the target classes for our classification: 1(seen), 2(uncertain), 3(not seen).

A bootstrapping technique with 20 iterations was used for performing learning and classification, this way the results are more stable and standard deviation can be computed. The instances in the dataset were not uniformly distributed. We shuffled the instances and then divided them in two separate sets for training and test, this way we assured that instances are uniformly distributed in both sets. For SVM and KNN 80 percent of the dataset was used for training and 20 percent for test.

The representation of the processing flow is presented in Figure 1. FFT was applied to EEG signal then we computed spectral power and we obtained the feature matrix. We used tree machine learning algorithms for classification and then we computed performance metrics.

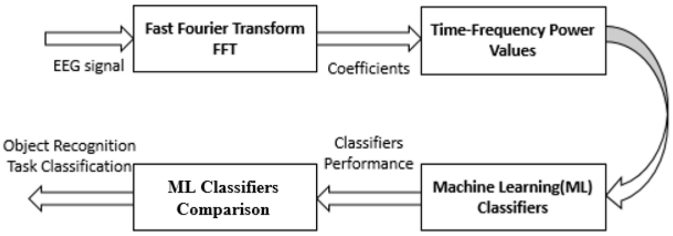

EEG signal
Fast Fourier Transform FFT
Coefficients
Time-Frequency Power Values
Machine Learning(ML) Classifiers
Classifiers Performance
ML Classifiers Comparison
Object Recognition Task Classification

Figure 1 Development flow

A. *Data format*

The data we used to construct our prediction model includes the EEG signal collected from the 128 electrodes and metadata information about events during recording: the events and the timestamp for each corresponding event. All files contain binary data: the format of the files with raw data recoded is 32-bit floating point values, the sampling frequency is 512Hz and the rest of files contain 32-bit integer values about the events that occur during the experiment. The following events are the most important: 150 marks the beginning of the baseline before each event, 129 occurs on stimulus presentation (end of baseline) and the next event after stimulus presentation is 1,2 or 3 representing the response of the subject (seen, not seen, uncertain). In order to compute the FFT power for a specific channel and trial we extracted the EEG raw data between event 129 (stimulus presentation) and 1,2 or 3 (response).

B. *Sliding window*

We used a sliding window for a better temporal precision while computing the FFT power. In the sliding window approach, a fixed length (usually smaller than interval length) window starting from the beginning of the interval is shifted with a step to the right on each iteration (see Figure 2Figure 3). The FFT power is computed in every iteration and in the end, the mean value is taken.

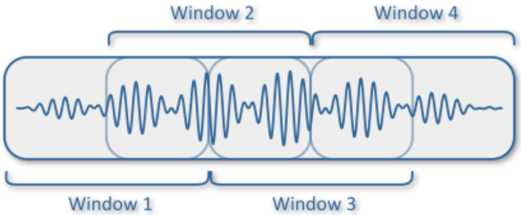


Figure 2 [2] Sliding window

The information about the size of the sliding window was mandatory for our results therefore, a sliding window tuning algorithm was created. The algorithm started with a window length equal to 128 samples and then with one bigger with 128 up to 3200 samples. The best results were obtained when the window was 1664 samples long, but because of the results with windows between 1500 and 3000 samples long were similar we chose a value of 2048 samples which is a good thing because FFT algorithm works faster when the window length is a power of 2.

C. *Feature extraction using FFT*

In our study, the FFT method was used to compute the power spectral density for the gamma frequency range. After extracting the EEG raw data between event 129 and the event outcome (1, 2 or 3) we applied a sliding window as described above. For each window during sliding we applied FFT to the time domain signal and we obtained the signal in the frequency domain. We computed the magnitude for each value of the signal in the frequency domain and then we took all magnitudes that were in the gamma frequency range and we averaged them. This has been done for each window during sliding and we averaged the partial results. The result of this process represents a feature in the feature matrix.

In Figure 3 is presented how we computed the feature matrix from the raw EEG data to the feature matrix.

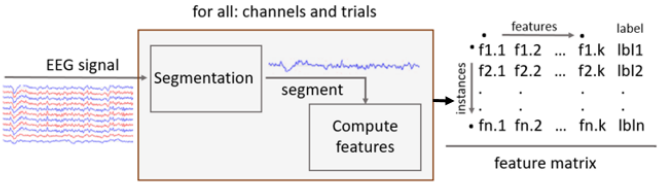


Figure 3 Feature extraction flow

D. *Choosing the right interval*

When FFT is computed for an array of length N, N/2 frequency bins are returned. Each bin contains spectral information in a range of frequencies. To compute the FFT power between two frequencies, all bins that are between those two values must be taken and then the power must be computed. Both Gamma and Beta high waves describe a subject that is highly connected to the outside world therefore, we created an algorithm to find the right frequency interval to compute the FFT mean where the interval boundaries and the interval length vary. Using this tuning we achieved our best results when the frequency interval for computing the FFT power was between 25Hz and 75Hz.

Another interesting thing is that this interval taken alone gave us better results than taking all waves together.

E. *Data splitting*

As described above, the same 30 objects were presented to subjects seven times, the image complexity increased in the end so the image could be recognized with less effort. So the instances in the dataset were not uniformly distributed and we have less instances of the *uncetain* (ii) class.

First 30 instances had their coresponding **g** value equals to 0.0 and so on until the last 30 instances had their 30 **g** equals to 0.3. We wanted these values to be uniformly distributed shuffled the dataset and then we split it in training and test set. We used a random, stratified 80%-20% split to generate train-test subsets.

F. *Zone tunning*

We also considered the main brain areas in order to identify the areas that are more involved in the object recognition task. As we know, our brain is divided in five big functional areas: Frontal, Parietal, Occipital, Left and Right Temporal lobes. The Left Temporal lobe is responsible for object recognition so we've decided to take the electrodes from a specific zone or different combinations of zones.

After this experiment, we found that the Parietal lobe had the lowest accuracy when each lobe was taken alone and even if the channels that were placed in the Parietal area were excluded, the accuracy remained the same. An important thing to be mentioned is that the Left Temporal and Right Temporal lobes together gave us accuracy with just 2% lower than best accuracy where all zones were considered.

IV. CLASSIFICATION METHODS

In our case, the feature is the time-frequency power and the response of the subject during the recording (seen, unseen, uncertain) defines the classes. The classifier is trained using the training set of data (for example 80% of the entire dataset for SVM and KNN and 70% for ANN) and it is tested using the remaining instances. To demonstrate the performance of the proposed feature extraction scheme in the object recognition task, we decided to investigate the SVM, KNN and ANN classifiers. For each classifier, we implemented both the binary and multiclass classification and for the testing part. Multiclass classification means that all tree classes were considered and in binary classification merged uncertain and unseen classes. Ee used the bootstrap method with 20 iterations.

A. *SVM*

Support Vector Machine (SVM) performs classification by finding a hyper-plane that maximizes the margin between two classes. SVM performs supervised learning. Using one-against-all method SVM can be used as a multi-class classification algorithm. In our approach, we rely on the implementation

[2] https://www.barryvan.com.au/files/Diag_SlidingWindow.png

defined in [8]. It models a given training set with a corresponding group vector and classifies a given test set using an SVM classifier according to a one vs. all relation.

B. KNN

K-Nearest Neighbors (KNN) is a simple and fast multi-class algorithm, where a case is classified by a majority vote of its neighbors.

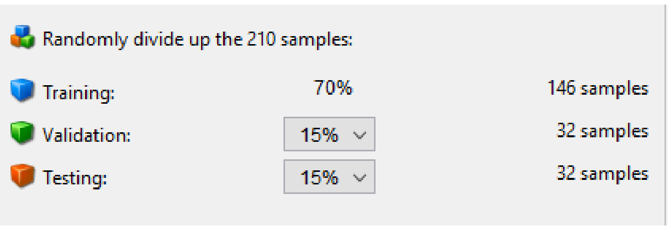
We achieved the best results for this particular problem using the following configuration of the matlab KNN algorithm:

- Distance: correlation
- NSMethod: exhaustive
- BreakTies: nearest
- NumNeighbors: 5

C. ANN

Artificial neural networks (ANNs) is a computational model based on the structure and functions of biological neural networks. The information that flows through the network affects the structure of the ANN because a neural network changes based on that input and output.

In our case, for the best results 70% from the dataset for training and 15%, 15% for validation and testing respectively a network with one hidden layer and 10 neurons were used and we used Levenberg-Marquardt as the training function.



Randomly divide up the 210 samples:

Training: 70% 146 samples

Validation: 15% 32 samples

Testing: 15% 32 samples

Figure 4 Data splitting for neural network learning

V. Results and Discussion

As we described above we have three possible labels for our data: 1(Seen), 2(unseen), 3(Uncertain). In one EEG experiment we have 210 instances of this tree classes and we used in our tests data that was recorded on 4 subjects, so we had 840 instances.

We obtained the best results using sliding window with window length: 2048 and step: 20, taking gamma power (25-75Hz) as only feature and using a random stratified 80%-20% split.

In this section, we call Multiclass: classification when all tree classes mentioned above are used and we call Binary: classification when uncertain instances are included in unseen instances.

For each binary and multiclass classification, we have 2 possibilities: Gamma: where the gamma power (28-75Hz) alone was taken (one value for each channel) and All bands: here we took 6 values for each channel (one for each wave but two for gamma: gamma low and gamma high). When we combine this two we got 4 possibilities:

- Gamma Multiclass
- Gamma Binary
- All bands Multiclass
- All bands Binary

A graphical representation of the results is shown in Figure 5. KNN was the algorithm that gave the best results for Gamma multiclass (the accuracy mean is 87%) and for Gamma Binary (accuracy mean is 91%). We know that KNN is sensitive to irrelevant features and this is the reason why introducing the spectral power of all frequency bands gave us an accuracy that is lower compared with the one obtained when only the gamma power was considered.

When all waves were considered, the SVM algorithm had better results in both cases (i.e. binary and multiclass). For multiclass classification, the accuracy mean for SVM is 79% and 84.4% for binary classification. Both SMV and KNN have standard deviations of accuracy below 3%. In almost all trials the accuracy mean of ANN was lower than SVM and KNN but unlike the other trials when all bands were used and multiclass classification was performed, the accuracy mean was very close to SVM and KNN. The average of the standard deviation is ~10% and this means that it is not very stable.

Usually we got poor performance for the uncertain class. For example, using KNN for the uncertain class we got 38% precision and 60% recall and using SVM we got 55% precision and 45% recall. We decided to combine uncertain instances with the other two classes: we combined them with the seen instances and we got 92.5% accuracy and 90.5% when they were combined with unseen using KNN. We concluded that the uncertain class is similar with both seen and unseen classes so we tested the case when the uncertain instances were excluded and we got 95.1% accuracy.

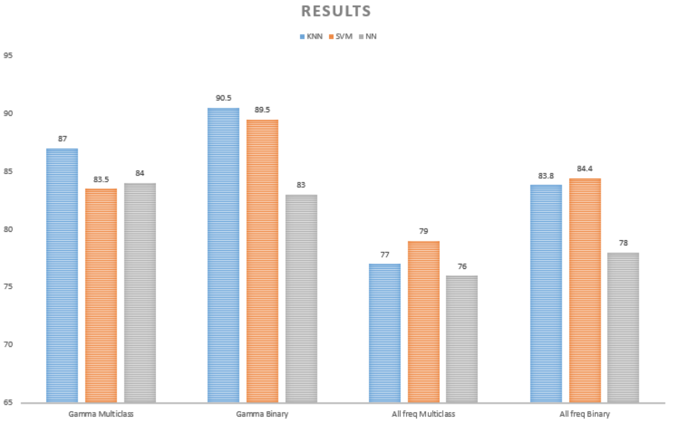


Figure 5 Accuracy for KNN, SVM and ANN

In the *Table 1* below, the classification mean accuracy and the standard deviation for each experiment and classifier is shown.

Table 1 Accuracy and Standard Deviation

Experiment	Accuracy / SD %		
	KNN	*SVM*	*ANN*
Gamma Multiclass	87 / 2.6	83.5 / 2.4	84 / 12

Experiment	**Accuracy / SD %**		
	KNN	***SVM***	***ANN***
Gamma Binary	90.5 / 2	89.5 / 2.55	83 / 8
All bands Multiclass	77 / 3.9	79 / 2.3	76 / 11
All bands Binary	84 / 2.9	84.4 / 3	78 / 10

VI. Conclusion

In this paper, we have presented the use of power spectral density computed using the fast Fourier transform during the presence of a visual stimuli along with machine learning algorithms used for classification of object recognition cognitive tasks. The gamma frequency range was used to compute the relative energies and, for classification, three different classifiers (SVM, KNN and ANN) were used and their performance (accuracy) was evaluated. During the presence of the stimuli the KNN classifier has an accuracy of 87% for multiclass (seen, not seen, uncertain) and if the not seen and uncertain labels are merged we obtain an accuracy of 95.1% (binary classification).

Acknowledgment

This research was supported by the Experimental and Theoretical Neuroscience Lab at Coneural Cluj-Napoca and the Technical University of Cluj-Napoca. Work supported by CNCS -UEFISCDI, grants PN-III-P4-ID-PCE-2016-0010 contract 78 /2017,

References

[1] V. K. Harpale and V. K. Bairagi, "Time and frequency domain analysis of EEG signals for seizure detection," 2016.

[2] H. U. Amin, A. S. Malik and R. F. Ahmad, "Feature extraction and classification for EEG signals using wavelet transform and machine learning techniques," 10 June 2015.

[3] S. P. Fitzgibbon, K. J. Pope, L. Mackenzie and C. R. Clark, "Cognitive tasks augment gamma EEG power," September 2004.

[4] I. T. L. M. R. C. M. Vasile V. Moca, "Visual Exploration and Object Recognition by Lattice," *PLoS ONE,* vol. 6, no. 7, pp. 1-16, 2011.

[5] A. Ciuparu and R. C. Muresan, "Sources of bias in single-trial normalization procedures," December 2015.

[6] Meditation Power, "Brainwaves," [Online]. Available: http://www.meditation-power.com/your-brainwaves.htm.

[7] P. Pramanick, "CLASSIFICATION OF ELECTROENCEPHALOGRAM (EEG) SIGNAL BASED ON FOURIER TRANSFORM AND NEURAL NETWORK".

[8] C. Neuburger, "Multi Class SVM".*https://www.mathworks.com/matlabcentral/fileexchange/39352-multi-class-svm.*