# O scurtă prezentare a metodelor de detectare a emoţiilor din semnalul vocal (Mai 2018)

Iordache-Stoicescu Laurenţiu-Iulian

***Abstract*— Object recognition plays a very important role in computer vision based applications. This type of application are vital for autonomous systems and for industrial applications. In this paper are presented some popular approaches of object recognition such as PCA, SIFT, SURF, FAST etc. Furthermore, the pros and cons of object recognition algorithms are described and a classification of those algorithms by the processing based method was attempted. Finally, these algorithms are compared in terms of accuracy and robustness.**

***Index Terms*— HMM – Hidden Markov Model SVM – Support vector machine LPC - Linear Predictor Coefficients MFCC - Mel-frequency Cepstrum Coefficients LFPC - Log-frequency power coefficients**

## I. Introducere

Semnalul vocal reprezintă cea mai rapidă metodă naturală de comunicare între oameni. Acest fapt a determinat apariția multor interfețe de comunicare om-mașină pentru a eficientiza metoda de interacțiune dintre om și mașină. În ciuda tuturor progreselor realizate în recunoașterea vorbirii, este încă grea realizarea unei interacțiuni naturale între om și mașină deoarece mașina nu pricepe emoțiile vorbitorului. Datorită acestei necesități a apărut ramura de recunoaștere a emoțiilor din vorbire, aceasta se ocupă cu analizarea stării emoționale a vorbitorului pentru a oferi diverse semantici mesajului transmis de către acesta.

### A. Necesitate

Recunoașterea emoțiilor din vorbire este folositoare în aplicații care necesită o interacțiune naturală între utilizator și mașină cu ar fi filme web și aplicații de învățare pentru care sunt necesare ca răspuns emoțiile utilizatorului. De asemenea, recunoașterea emoțiilor mai sunt importante și în aplicații de bord ale autovehiculelor în care informația legată de starea mentală a șoferului poate fi utilizată în favoare siguranței acestuia. O altă aplicație în care se poate utiliza recunoașterea emoțiilor o reprezintă sistemele de traducere automată în care simpla stare a utilizatorului poate varia înțelesul traducerii.

### B. Generalități

Sarcina de detecție a emoțiilor este o sarcină complicat de realizat. Nu este clar ce caracteristici ale vorbirii sunt importante pentru distingerea emoțiilor. Pe lângă această incertitudine se mai adaugă și variabilitatea semnalului introdusă de diversitatea propozițiilor, vorbitorilor, ciclurilor de vorbire și a vitezelor de vorbire. Această sarcină mai este îngreunată și de faptul că de cele mai multe ori nu este prezentă o singură emoție la un vorbitor, acestea există cumulativ în enunțurile rostite deoarece. Încă un motiv este acela că manifestarea emoțiilor variază de la de la un vorbitor la altul și pe baza culturii și a mediului acestuia.

Emoțiile nu au o definiție teoretică ce este acceptată de toată lumea. Este foarte întâlnit ca acestea să fie caracterizate în două tipuri:

*1) De activare:* Activarea se referă la cantitatea de energie necesară pentru a exprima o anumită emoție.

Anumite emoții cum ar fi bucuria, furia și frica provoacă o creștere a pulsului, a tensiunii, schimbări ale ritmului respirației, presiune subglotală mărită, uscare a gurii și ocazional tremurări musculare. Toate aceste modificări fiziologice provoacă schimbări în caracteristicile acustice ale mesajului transmis. Mesajul este enunțat rapid cu sonor ridicat și o frecvență audio ridicată.

Alte emoții cum ar fi tristețea induc efecte inverse față de cele anterior menționate și anume scăderea pulsului și a tensiunii, creșterea gradului de umiditate a gurii. Aceste emoții determină un mesaj audio ce este rostit la intensitate scăzută, cu mai puține frecvențe înalte.

*2) De valență:* De valență: Emoțiile nu pot fi caracterizate doar având în vedere nivelul de energie necesar. De exemplu, bucuria și furia sunt ambele emoții cu un nivel mare de energie dar afectează diferit. Această diferență este caracterizată de dimensiunea valenței. Din nefericire nu se știe precis cum această valență afectează caracteristicile acustice pentru alte emoții. De aceea, clasificarea între emoțiile de excitare înaltă și cele de excitare joasă se poate realiza cu precizie ridicată iar clasificarea dintre diferite alte emoții este încă greu de realizat.

O altă chestiune importantă în recunoașterea emoțiilor constă în alegerea unui set definit de emoții. Lingviștii au definit un set de aproximativ 300 de emoții pe care le întâlnim pe parcursul vieții. Acest pachet de emoții este însă prea mare pentru a putea fi utilizat momentan, de aceea se recurge la utilizarea unui set restrâns de emoții de bază. O idee foarte interesantă este aceea că emoțiile pot fi descompuse în aceste emoții de bază precum culorile pot fi descompuse în culori primare. Emoțiile primare sunt definite de psihologul Paul Ekman ca fiind: tristețe, agonie, furie, surprindere, frică, dezgust și bucurie.

### C. Baza de Date

O chestiune importantă de luat în calcul în evaluarea unui sistem de recunoaștere a emoțiilor din vorbire este gradul de naturalețe al bazei de date utilizată în evaluarea performanțelor. Se pot trage concluzii incorecte despre sistem dacă datele utilizate nu sunt de calitate.

Din păcate, majoritatea bazelor de date cu date audio ce conțin înregistrări preluate de la vorbitori cu anumite emoții nu sunt disponibile public. Acest fapt determină o îngreunare a procesului de evoluție în această ramură. Cercetătorii nu sunt capabili sa se coordoneze din punctul de vedere al greșelilor și le repetă pentru baze de date diferite.

### D. Fapt

În carlinga avioanelor s-a descoperit că sistemele de recunoaștere a vorbirii antrenate cu mesaje de la un subiect aflat sub stres au performanțe mult mai bune decât sistemele antrenate cu mesaje de la subiecți aflați în stare normală.

## II. Caracteristici utilizate pentru recunoașterea emoțiilor

Pentru realizarea unui sistem de recunoaștere a emoțiilor din vorbirea unui subiect este necesară identificarea caracteristicilor

ce caracterizează diversele emoții. O alegerea caracteristicilor va influența performanțele sistemului.

Pentru o analiză eficientă a caracteristicilor trebuie să se țină seama de regiunea de analiză utilizată pentru extragerea caracteristicilor, să se aleagă caracteristica potrivită pentru identificarea emoției, să se ia în calcul efectele proceselor uzuale de procesare a semnalului vocal cum ar fi filtrarea și eliminarea zonelor de liniște dar și de asemenea să se ia în calcul faptul că trăsăturile acustice s-ar putea să nu fie suficiente pentru modelarea emoțiilor și că ar putea să se folosească și alte tipuri de caracteristici cum ar fi informația discursului sau trăsăturile faciale. În continuare vor fi detaliate aceste aspecte.

### A. Caracteristici locale și caracteristici globale

Primul aspect se referă la selectarea regiunii de analiză folosită pentru extragerea caracteristicilor. Aici există două abordări. Divizarea semnalului în cadre de durată scurtă pentru care se calculează vectorul caracteristic sau extragerea de mărimi statistice globale pentru întreaga propoziție.

Deoarece semnalul vocal este nestaționar, se practică divizarea acestuia în ferestre mici de 20 până la 40 de ms. O caracteristică importantă a semnalului vocal, care facilitează analiza acestuia este că acesta este aproximativ staționar pe perioade scurte de timp. Din fiecare astfel de fereastră sunt extrase caracteristici prozodice cum ar fi frecvența fundamentală și energia și denumite caracteristici locale.

Pe de altă parte, caracteristicile globale reprezintă statistici ale tuturor caracteristicilor locale extrase din mesaj. Majoritatea cercetătorilor au agreat faptul că aceste caracteristici globale sunt superioare caracteristicilor locale din punct de vedere al preciziei și a timpului de clasificare. Caracteristicile globale mai au încă un avantaj față de cele locale și anume acela că sunt mult mai puține la număr, ceea ce facilitează o execuție mult mai rapidă a unor algoritmi cum ar fi cel de ”validare în cruce”. Algoritmul de validare în cruce reprezintă o tehnică de evaluare a modelelor predictive petiționând mostra originală într-un set de antrenare pentru antrenarea modelului și un set de evaluare [2]. Utilizarea acestora este eficientă doar în cazul distingerii emoțiilor de excitație înaltă (ex. furia, frica și bucuria) de cele de excitare joasă(ex. tristețea). Caracteristicile globale nu reușesc să realizeze o clasificare precisă a emoțiilor cu excitare similară de exemplu identificarea furiei față de bucurie. De asemenea, mai prezintă și dezavantajul că informația temporală prezentă în mesaj este pierdută. Și încă un dezavantaj îl reprezintă chiar faptul că sunt într-un număr mai mic decât caracteristicile locale deoarece s-ar putea să nu se poată antrena modele de tipul HMM sau SVM.

O altă abordare pentru extragerea de caracteristici o reprezintă segmentarea semnalului pe durate scurte pentru analiza fonemelor. Această abordare se bazează pe faptul că forma spectrului fonemelor variază în funcție de emoții. Această afirmație este valabilă doar pentru vocale.

### B. Clasificarea caracteristicilor

Un aspect important în recunoașterea emoțiilor îl reprezintă extragerea caracteristicilor semnalului vocal care caracterizează eficient emoția din mesaj dar în același timp este independentă de vorbitor sau de conținutul lexical.

Caracteristicile vorbirii pot fi grupate în următoarele categorii: Caracteristici continue, caracteristici calitative, caracteristici spectrale și caracteristici bazate pe operatorul TEO. În figura 1 se pot observa exemple de caracteristici aparținând fiecărei categorii.

*1) Caracteristici de vorbire continue:* Caracteristicile prozodice continue cum ar fi frecvența fundamentală și energia exprimă o mare parte din sentimentele existente într-un mesaj. S-a observat că starea de excitare a subiectului influențează distribuția spectrală a energiei

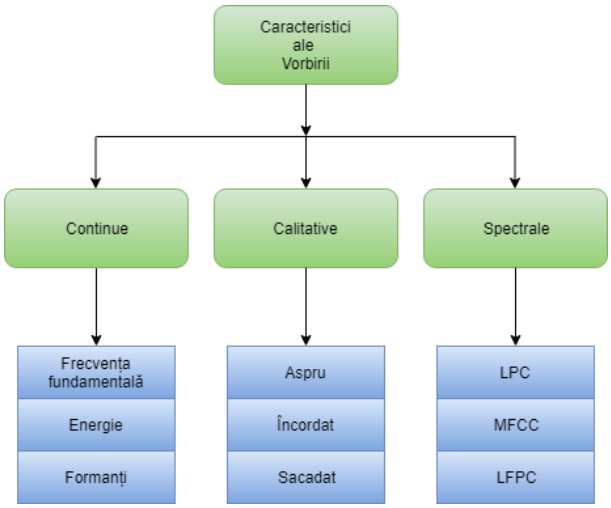


Fig. 1. Categorii de caracteristici ale semnalelor vocale

precum și frecvența și durata pauzelor din vorbire. Conform studiilor, aceste caracteristici acustice pot fi grupate în [1]:

- Caracteristici asociate frecvenței fundamentale;
- Caracteristici bazate pe formanți
- Caracteristici asociate energiei semnalului vocal
- Caracteristici ale articulării

*2) Caracteristici de calitate a vocii:* Calitatea vocii este strâns legată de tipurile de emoții, aceasta este descrisă de emoții care direcționează puternic subiecții către un șir de acțiuni. Acestea sunt opuse emoțiilor fundamentale care influențează pozitiv sau negativ acțiunile și gândurile unei persoane. O gamă largă de variabile fonetice contribuie la impresia subiectivă de calitate a vocii. Corelațiile acustice sunt grupate în următoarele categorii [1]:

- Nivelul vocii, amplitudinea semnalului, energia și durata s-au dovedit a fi metrici bune pentru nivelul vocii.
- Frecvența fundamentală a vocii
- Fraze, foneme, cuvinte
- Structurile temporale

*3) Caracteristici bazate pe spectru:* Acestea sunt adesea reprezentări ale semnalului pe perioade scurte de timp. Emoțiile dintr-un mesaj influențează distribuția spectrală de energie a mesajului. De exemplu, mesajele care sunt rostite când subiectul este fericit au un nivel ridicat al energiei pentru frecvențele înalte pe când cele rostite de subiecți care sunt triști emoțional au nivele scăzute ale energiei pentru aceleași frecvențe. Pentru o mai bună exploatare a gamei de frecvențe, spectrul este trecut printr-o serie de filtre trece-bandă. Caracteristicile spectrale sunt extrase pe baza rezultatelor obținute de la fiecare filtru. Deoarece percepția umană a frecvenței fundamentale nu este modelată liniar, filtrele trece-bandă sunt distribuite uniform cu privire la o metodă neliniară potrivită cum ar fi scala frecvențelor Mel.

### C. Procesarea vorbirii

Înaintea extragerii caracteristicilor este necesară o preprocesare a semnalului audio. De exemplu, datorită diferențelor din mediul de înregistrare este necesară o normalizare a energiei pentru toate mesajele. Alt exemplu îl reprezintă netezirea contururilor extrase, pentru aceasta se folosește metoda suprapunerii cadrelor. Pentru eliminarea ondulațiilor din spectrul de frecvențe sunt utilizate ferestre de tip Hamming.

Deoarece intervalele de liniște pot oferi informații importante asupra stării emoționale, acestea sunt păstrate intacte, față de procesele normale de analiză a vorbirii.

Odată extrase caracteristicile se poate să fie necesară o postprocesare înainte de a antrena clasificatorul. De exemplu, este posibil ca vectorii extrași să aibă unități diferite și prin urmare, valorile lor numerice pot avea ordine de mărime diferite sau pot lipsi cu desăvârșire.

A
A
A
A
In order to find features, the eigenvalues are computed for each pixel. Constructing the response map can be done by calculating the cornernes measure $C(x, y)$ for each pixel $(x, y)$ using the

$$C(x, y) = det(M) - K(trace(M))^2 \quad (1)$$

where

$$det(M) = \lambda_1 * \lambda_2, and trace(M) = \lambda_1 + \lambda_2 \quad (2)$$

The K is an adjusting parameter and $\lambda_1, \lambda_2$ are the eigenvalues of the auto-correlation matrix. The process of computing the eigenvalues is computational expensive because of the square root. Harris suggested using this cornerness measure that combines the two eigenvalues in a single measure. The non-maximum suppression should be done to find local maxima and all non-zero points remaining in the cornerness map are the searched corners.

*1) FAST Feature Detector:* Features for accelerated segment test(FAST) is a single-scale corner detection method which is used to extract feature points. The FAST corner detector was originally developed by Edward Rosten and Tom Drummond and was published in 2006. The biggest advantage of FAST is, as the abbreviation of it's name says it, that is a fast processing tool.

The process of corner detection consist by applying a segment test to each pixel by considering a sampling circle of 16 pixels around the corner candidate pixel as a base of computation. If a set of $n$ neighboring pixels in the Bresenham circle with the radius $r$ are all brighter or darker than the candidate pixel plus a threshold value $t$, then the considered pixel is classified as a corner as exemplified in figure 3.

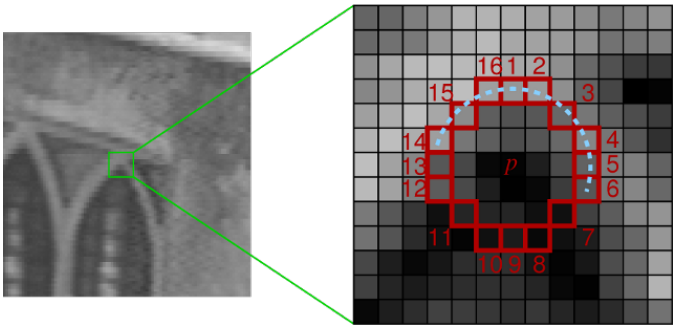


Fig. 2. Corner detection using Bremenham sampling circle applied to a corner candidate.

A high-speed test to exclude all non-corners points is used. In this test only pixels 1, 5, 9 and 13 are examined. A corner can exist only if three of those pixels are brighter or darker than the considered pixel $p$. Based on this statement, the computation time is enhanced by fast eliminating the pixels that are not corners. Although the high speed test yields high performance, it has several weakness. An improvement for these limitations and weakness is achieved using a machine learning approach. The ordering of questions used to classify a pixel is learned by using the "well-known decision tree algorithm (ID3)", which speeds this step up significantly [2].

*2) Laplacian of Gaussian:* Laplacian of Gaussian (LoG) is a common blob detector. The Laplacian is a 2D isotropic measure of the second spatial derivative of an image used to find areas of rapid change in images (edges). Since derivative filters are very sensitive to noise, it is common to smooth the image using a Gaussian filter before applying the Laplacian. This two-step process is called the Laplacian of Gaussian operation.

Given an input image $I(x, y)$, the scale space representation of the image is defined by $R(x, y, \sigma)$ is obtained by convolving the image with a variable Gaussian kernel $G(x, y, \sigma)$ where

$$R(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (3)$$

and

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{\frac{-(x^2+y^2)}{2\sigma^2}} \quad (4)$$

To include a smoothing Gaussian filter, combine the laplacian and the Gaussian functions to obtain a single equation/ The LoG operator takes the second derivative of the image. The LoG will give zero where the image is uniform and where changes occurs, the Log will give a positive response on the darker side and a negative response on the lighter side.

$$LoG(x, y) = \nabla^2 G(x, y, \sigma) - \frac{1}{\pi\sigma^4}(1 - \frac{x^2 + y^2}{2\sigma^2}) e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (5)$$

However, the operator response is strongly dependent on the relationship between the size of the blob structure and the size of the smoothing Gaussian kernel [2]. The scale control is achieved by changing the standard deviation $\sigma$. The computation of LoG operators is time consuming so, as an alternative to this method is used Difference of Gaussian (DoG).

*3) Difference of Gaussian:* Difference of Gaussians (DoG) represents a feature enhancement algorithm that involves the subtraction of one blurred version of an original image from another one less blurred. The blurred images are created as in the Laplacian of Gaussian algorithm by convolving the original image with a variable Gaussian kernel.

Blurring an image with a Gaussian kernel suppresses only high-frequency spatial information. Subtracting one image from another preserves spatial information that lies between the range of frequencie that are preserved in the two blurred images. Thus, the DoG is a band-pass filter that discards all but a handful of spatial frequencies that are present in the original image. This approach is used in scale-invariant transform (SIFT) algorithm. In this context, the DoG gives a close approximation to the Laplacian of Gaussian and is more computational friendly because it can be computed without convolution.

## D. Feature Descriptors

After detecting a interest point from an image at a certain location $p(x, y)$, the neighborhood of p needs to be described in a suitable manner in order to be discriminative and insensitive to local image deformations. In general, the problem we are focusing on is that of comparing two image patches and measuring their similarity. There are a large number of image feature descriptors in the literature.

Further will be presented some frequently used descriptors.

*1) Scale Invariant Feature Transform:* Scale Invariant Feature Transform (SIFT) is an algorithm used to describe keypoints detected with int the image using the DoG operator. For each interest point, a feature vector is extracted. The algorithm computes the orientation of the image over a number of scales and over a patch around the interest point in order to provide rotation invariance.

The SIFT descriptor builds a histogram of gradient magnitudes and orientations for a 16 x 16 pixels region around each interest point using its scale to select the level of Gaussian blur for the image. After, a set of orientation histograms with samples from a 4 x 4 subregion of the original $16x16$ patch is created. Each one have eight orientations bins [4].

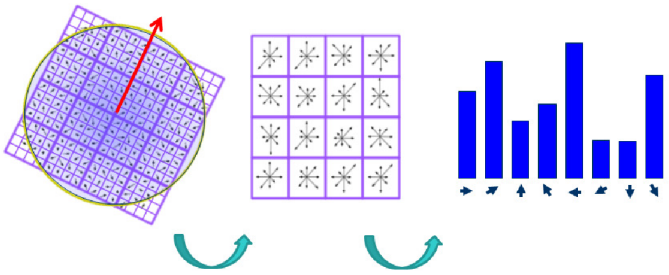

Figure 6. A keypoint descriptor

Fig. 3. Representation of SIFT descriptor for a 16 x 16 pixel patch weighted by a Gaussian falloff indicated by overlaid circle.

The magnitude of each point is wighted with a Gaussian weighting function in order to give and less weight to gradients far from the interest point. There fore, feature vector dimension is 128 (4 x 4 x 8). Finally, the feature vector is normalized to unit length. The normalization offers invariance to affine changes and illumination. Some changes can occur due to camera saturation or similar effects. A good way to combat these effects is to threshold the values in the feature vector to a maxim value of 0.2 [3].

The standard SIFT descriptor has the following advantages: it avoid problems du to boundary effects-smooth changes in location thanks to the representation of characteristics, is fairly compact thanks to the 128 elements vector, orientation and scale don't cause big changes in the descriptor vector. These characteristics are evidenced in excellent matching performance under different scales, rotations and lighting. It's main disadvantage is the processing speed due to high dimensionality.

*2) Speed-Up Robust Features Descriptor:* Speed-Up Robust Features Descriptor (SURF) is an alternative to SIFT. This algorithm presents a better computational speed and a better robustness while keeping the advantages of SIFT.

The feature detection is based on simple 2D filters, it uses a scale invariant blob detector based on the determinant of Hessian matrix for both scale selection and location [2]. While SIFT uses approximated LoG for finding scale-spaces, SURF approximates LoG with Box Filter. One big advantage of this approach is that, convolving with box filters can be done with the help of integral images and it can be done in parallel for different scales.

The SURF descriptor begins by constructing a square region centered on the detected interest point ant orientated along its main orientation. Th size of the window is $20s$ with $s$ equals to the scale at which the feature is detected. After considering a window, is divided into smaller 4 x 4 sub-windows and for each one Haar wavelet response in the vertical and horizontal directions are computed at a 5 x 5 sampled points as shown in Figure 6.

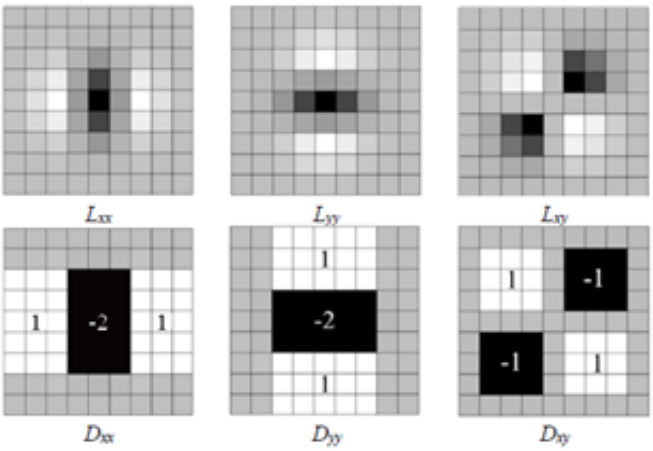


Fig. 4. Representation of SIFT descriptor for a $\mathbf{16x16}$ pixel patch weighted by a Gaussian falloff indicated by overlaid circle.

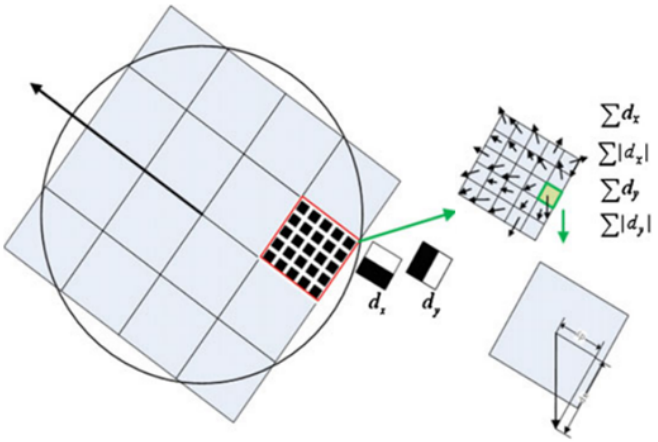


Fig. 5. Haar wavelet response sampling weighted by a Gaussian falloff indicated by overlaid circle and orientation computing.

The wavelet responses are weighted with a Gaussian window centered on the interest point in order to increase the robustness against geometric deformations and errors generated by localization. The responses $d_x$ and $d_y$ are summed up for each sub-window and inserted in a feature vector $v$ where

$$v = (\sum d_x, \sum |d_x|, \sum d_y, \sum |d_y|) \tag{6}$$

The feature vector is computed for each sub-window, resulting a feature descriptor of length 4 x 4 x 4 = 64 dimensions. The feature descriptor is further normalized to a unit vector in order to reduce illumination effects [5].

The main advantage of SURF descriptor compared to SIFT is the processing speed. This is given by the feature vector's dimension, it uses only 64 values in order to describe an interest point.

Beside the disadvantage of computational complexity of both SURF and SIFT descriptors they have another one that may affect the decision to choose one of them, they are patented by the institutions so in order to use them you must pay.

*3) Binary descriptors:* Considering the fact that SIFT and SURF descriptors are based on HoG which implies computing gradients for every pixel from the patch surrounding the interest point, they are costly from the perspective of time and processing resources. This is where binary descriptors come in handy.

The principle on which they are based is to encode most of the information of a patch just by comparing intensity images. This can be done very fast only using the Hamming distance as a distance measure between two binary vectors and then matching between two patch descriptions. The matching process can be done very easy just by computing the sum of the XOR operation between two binary strings.

In general, binary descriptors have three constituent parts: A sampling pattern, orientation compensation algorithm and a sampling pairs pattern. In order to describe a patch is needed a sampling pattern of a window of various size centered on the interest point.

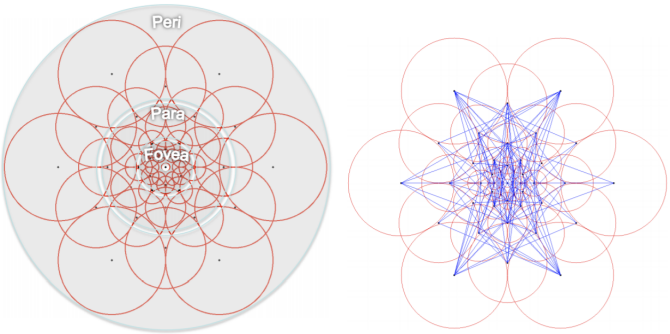


Fig. 6. FREAK sampling pattern and sample pairs generating algorithm [7].

With the resulted samples from the patch the algorithms consider a number of pairs of points. For each pair the intensity value is compared. If the first value is bigger it will be represented as '1' in the description vector else will be represented as a '0'. In order to compare two patches it is enough just to count the number of bits where description vector differs. This is done applying

$$v = \sum (d_x \oplus d_y) \tag{7}$$

where $d_x$ is the description vector of the known interest point $d_y$ is the description vector of the supposed point in another image. Each binary descriptor has its own sampling patterns, pairs creating patterns and orientation deduction algorithms. Further will be briefly presented some binary algorithms particularities.

*a) BRIEF:* BRIEF was the first binary descriptor published. It doesn't have any orientation algorithm or elaborated sampling pattern. BRIEF take only the information from single pixels in order to build the descriptor so in order to be less sensitive to noise the image must be smoothed first.

In order to build the descriptor it must define a sampling pattern in order to determine $n$ pairs $(x_i, y_i)$. The authors consider five methods to determine these vectors [8] :

- $X$ and $Y$ are uniform distributed over the patch.
- Locations are Gaussian sampled, this means that the center region has more samples that the marginal areas.
- $X$ and $Y$ are sampled using two distinctive Gaussian sampling patterns that vary in standard deviations. This forces the samples to be more local.
- $X$ and $Y$ are randomly sampled from discrete location of a coarse polar gird.
- For each pair, the $X$ member will be the interest point and $Y$ takes all possible values on a coarse polar grid centered on the interest point.

As with all binary descriptors, computing the description vector consists in comparing the two pairs of samples. The process of matching two interest point is the same, just using the XOR operation between the two description vectors.

*b) ORB:* ORB descriptor is a bit similar to BRIEF, it doesn't have an elaborate sampling pattern ass BRISK or FREAK. However, there are two main differences between ORB and BRIEF:

- ORB make use of a orientation compensation mechanism, this confers it rotation invariant.
- ORB auto-adapts the sampling pairs in order to obtain the optimal one.

ORB is basically a fusion of FAST keypoint detector and BRIEF descriptor with many modifications to enhance the performance. First it use FAST to find keypoints, then apply Harris corner measure to find top N points among them. It also use pyramid to produce multiscale-features. But one problem is that, FAST doesn't compute the orientation [9].

It computes the intensity weighted centroid of the patch with located corner at center. The direction of the vector from this corner point to centroid gives the orientation. To improve the rotation invariance, moments are computed with x and y which should be in a circular region of radius r, where r is the size of the patch [9].

*c) BRISK:* The BRISK descriptor is different from BRIEF and ORB by having a defined sampling pattern. This consists in concentric sampling circles. For each sampling circle we apply Gaussian smoothing in order to reduce the noise sensibility. The standard deviation of each circle vary in depending on how close is from the center.

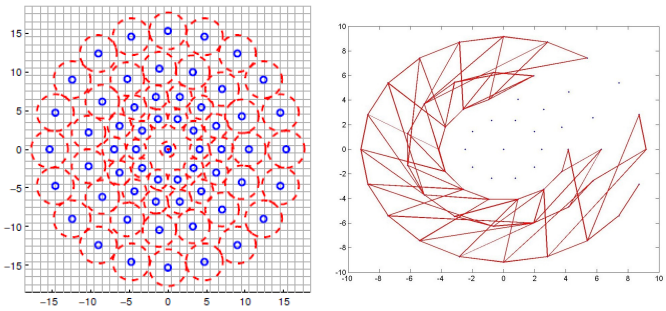

Fig. 7. BRISK sampling pattern and sample pairs generating algorithm [6].

When creating the pairs of samples we distinguish short distance pairs and long distance pairs. Short pairs are pairs of sampling that are used for the intensity comparisons that build the descriptor and long pairs are used to compute the orientation of the patch.

*d) FREAK:* FREAK is a descriptor with a handcrafted sampling pattern that uses machine learning techniques to learn the optimal set of sampling pairs [6].

Many sampling patterns are possible to compare pixel intensities. As we have seen, BRIEF uses random pairs, ORB uses learned pairs and BRISK uses a circular pattern where points are equally spaced on circles concentric.FREAK suggests to use the retinal sampling grid which is also circular with the difference of having higher density of points near the center. The density of points drops exponentially to the extremities as can be seen in Figure 7. Each sample point is smoothed with a Gaussian with the standard deviation given by the radius of the corresponding circle.

A possible strategy of choosing sampling points is to follow the ORB approach and try to learn the pairs by maximizing variance of

the pairs and taking pairs that are not correlated in order to maximize the sampled information. One of the many sampling patterns possible is illustrated in figure 7.

FREAKS takes advantage of this coarse-to-fine structure to speed up the matching using a cascade approach. When matching two descriptors, we first compare only the first 128 bits. If the distance is smaller than a selected threshold, we further continue the comparison until the next 128 bits. As a result, a cascade of comparisons is performed accelerating even further the matching as more than 90% of the candidates are discarded with the first 128 bits of the descriptor [6].

In order to compensate rotation changes, FREAK measures the orientation of the interest point by using a predefined set of 45 long distance symmetric sampling pairs similarly to BRISK descriptor.

TABLE I
COMPARISON OF THE FOUR BINARY DESCRIPTORS

| Name | Sampling pattern | Orientation calculation | Sampling pairs |
|---|---|---|---|
| BRIEF | None | None | Random |
| ORB | None | Moments | Learned pairs |
| BRISK | Concentric circles with more points on outer rings | Comparing gradients of long pairs | Using only short pairs |
| FREAK | Overlapping Concentric circles with more points on inner rings | Comparing gradients of preselected 45 pairs | Learned pairs |

### E. Features Matching

Features matching or generally image matching is the task of establishing correspondences between two images of the same scene. The matching is achieved by using the detected feature position (keypoint) and its descriptor.

A common approach to image matching consist in detecting a set of interest points, each associated with image descriptors of the image data. Once the features and descriptors are computed from two or more images the next step consist in establishing some preliminary feature matches between these images as shown in Figure 9.

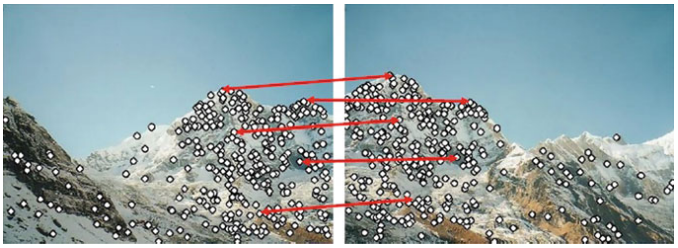

Fig. 8. Matching image regions based on their local feature descriptors [6].

The problem of image matching can be solved as follows. Suppose that p is detected feature point with a feature detector associated with a descriptor

$$\Phi(p) = \{\phi_k(P) \quad | \quad k = 1, 2, ..., K\} \tag{8}$$

where, for all K, the feature vector provided by provided by the $k$-th descriptor is

$$\phi_k(p) = (f^l_{1p}, f^k_{2p}, ..., f^k_{n_k p}) \tag{9}$$

The task is to find the best correspondence $q$ in the other images from the set of $N$ interest points $Q = \{q_1, q_2, ..., q_N\}$ by comparing the description vector $\phi_k(p)$ with those of the points in the set $Q$.A distance measure between the two interest points descriptors $\phi_k(p)$ and $\phi_k(q)$ can be defined as

$$d_k(p, q) = |\phi_k(p) - \phi_k(q)| \tag{10}$$

Based on the distance $d_k$, the interest points form the other images $Q$ are sorted in ascending order independently for each descriptor. A match between the pair of interest point $(p, q)$ is considered only if $p$ is the best match for q in relation to all the other points in the first image and $q$ is the best match for p in relation to all the other points in the second image [2]. To match vector based features we can use the nearest-neighbor matching in the feature space of the image descriptors in Euclidean norm. In order to reduce the ambiguity of the of matching candidates correspondence, the ratio between the distances of the nearest and the next nearest image descriptor must be less than a given threshold. Two algorithms have been found to be the most efficient: the randomized k-d forest and the fast library for approximate nearest neighbors (FLANN) [2].

On the other hand, these algorithms are not suitable for binary descriptors. Binary features are matched using Hamming distance computed with a bitwise XOR operation followed by a bit count on the result after determining the orientation of the patch and rotating it to match the orientation of the searched one.

The performance of matching methods based on interest points and descriptors depends on both the properties of the interest point and the choice of feature descriptor. The detectors and descriptors should be selected based on the image contents and application types. For instance, for a application that shall detect and track blobs it must use a blob detector not a corner detector and vice versa.

## III. ARTIFICIAL NEURAL NETWORKS

An Artificial Neural Network (ANN) is an information processing model that is inspired by the way biological nervous systems process information, such as the brain. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems.

Neural networks can be used to extract patterns and detect trends that are too complicated to be noticed by either humans or other computer techniques. A trained neural network can be thought of as an "expert" in the category of information it has been given to analyze because we feed him with a lot of information from the areas of interest.

Other advantages include:

- Adaptive learning: ability to learn how to do tasks based on the data given for training or initial experience;
- Self-Organizing: the capability of organizing or representing the information it receives during learning time in a useful manner;
- Real Time Operation: computations may be carried out in parallel, and special hardware devices are being designed and manufactured which take advantage of this capability;
- Fault Tolerance via Redundant Information Coding: Partial destruction of a network leads to the corresponding degradation of performance. However, some network capabilities may be retained even with major network damage.

### A. Human Object recognition

Much of the power of artificial intelligence stems from cloning the human behavior. One attempt to understand human object recognition ist the theory of $RecognitionByComponents$. The basis for this theory is that human primary way of classifying objects is by identifying their components and the relational properties among these components, rather than by features as texture or color. Recognition

is based on decomposition of an object in geons and utilizes the relation among entities, in the form of orientation, distance, size and connections.

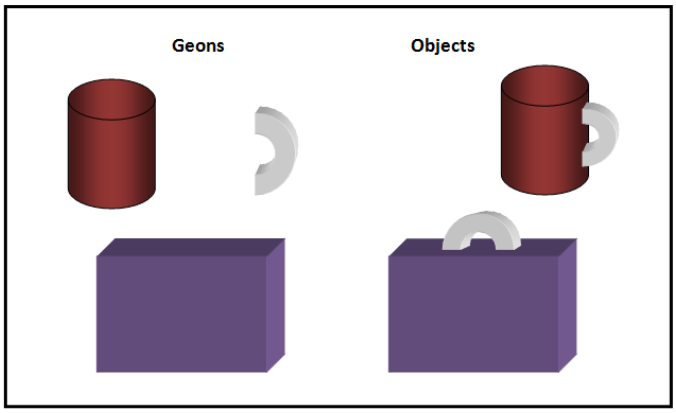


Fig. 9. A set of geons and some of the objects that can be composed by them [10].

In order to find evidences for this theory, Biederman tested human recognition system, mainly by using response time as empirical evidence. Hist tests shown that that varied visibility of geons and their connections clearly influence the way that the brain process the visual information. Main conclusions drawn in Biederman's research are [10]:

- Recognition is based on decomposition of an object in geons;
- Recognition is more likely to be based on shape information than on color and texture;
- Recognition utilizes the relation among entities, in the form of orientation, distance, size and connections;
- Intersections of lines making up geons are important for recognition, more so than having long untouched lines fully intact.

## B. Preprocessing

The goal of the preprocessing step is to extract the shapes present in the image, which is performed in three steps. First we use an edge detection algorithm to find the edge pixels in the image. A second step uses the generated edge pixels to create the contours of the shapes and in final step the resulting image of the second step will be used to extract the shapes that have now become visible.

The main motivation for the recognition by shapes is that in recognition of an object as a whole every discrepancy, even if it is only a small one, can disrupt the recognition process. As such, trying to recognize an object as a whole is possible as long as the object is not partly covered, or seen from a point of view that exposes a different side of the object than the system has been trained for. To make sure recognition can also take place in less perfect situations the shapes can be rotated to a generalized position so that the number of possible projections of the shape onto the network is reduced to a subset of the possibilities present when a shape can appear in all of the different positions.

To further reduce the number of positions the shape can be in, we can also attempt to remedy the mirroring possibility. Often just small parts can indicate the presence of an object. For example, just the presence of a hand can indicate that there's a person present or a set of wheels can give reason to believe a car might be in the scene.

## C. Representation

After the preprocessing step, it will be created the representations of the shapes for feeding them into a neural network. These representations can be interpreted as shape descriptors or, just descriptors. A shape descriptor is a set of numeric values that describes the shape in a way that makes it distinguishable from other shapes. It is less sensitive to scaling or rotation of the shapes, so that a shape that is upside down can result in the same representation as a shape in normal position.

## D. Interpretation

Having a set of descriptor vectors of which each is about a hundred values long is a difficult basis for object recognition. This is why the interpretation network should facilitate a reduction in this dimension, by somehow lowering the number of values to a short indication or description of the shape. The network will therefore take the descriptor of a shape as an input and output the classification of the shape.

## E. Learning process

In order to train a neural network it would be a possibility to highlight every local shape of an object that is equal to a shape from another identical one, this would lead to a great number of possible shapes, while the scope is to reduce this number as far as possible.

*1) Associative mapping:* Associative mapping in which the network learns to produce a particular pattern on the set of input units whenever another particular pattern is applied on the set of input units. This can be broken down into another two mechanisms:

*a) auto-association:* an input pattern is associated with itself and the states of input and output units coincide. This is used to provide pattern completion, to produce a pattern whenever a portion of it or a distorted pattern is presented. In the second case, the network stores pairs of patterns building an association between two sets of patterns.

*b) hetero-association:* is related to two recall mechanisms:

- nearest-neighbor recall, where the output pattern produced corresponds to the input pattern stored, which is closest to the pattern presented.
- interpolating recall, where the output pattern is a similarity dependent interpolation of the patterns stored corresponding to the pattern presented. Yet another paradigm, which is a variant associative mapping is classification, when there is a fixed set of categories into which the input patterns are to be classified.

*2) Regularity detection:* Regularity detection in which units learn to respond to particular properties of the input patterns. Whereas in associative mapping the network stores the relationships among patterns, in regularity detection the response of each unit has a particular 'meaning'. This type of learning mechanism is essential for feature discovery and knowledge representation.

Modifying the knowledge stored in the network as a function of experience implies a learning rule for changing the values of the weights. Information is stored in the weight matrix W of a neural network. Learning is the determination of the weights. Following the way learning is performed, we can distinguish two major categories of neural networks:

- fixed networks in which the weights cannot be changed, ie $\frac{\partial W}{\partial t} = 0$. In such networks, the weights are fixed apriori according to the problem to solve.
- adaptive networks which are able to change their weights, ie $\frac{\partial W}{\partial t} \neq 0$.

A first option is to use evolutionary techniques to find the best network. This option would be slightly preferred over random initialization, because it would work with several random networks and

will enable us to retrieve the best of all their properties. A second option is to set the weights of the network randomly and leave them fixed. Although this process can lead to very different results, as soon as a network performs well it can be stored and used in all future cases. Another option was to come up with a set of primitive properties that are inherent to all shapes, classify a large set of many different shapes to these properties, and train the network to learn to indicate the presence or absence of these properties.

The objective is to create a system that is as autonomous and adaptable as possible. This excludes the actual labeling and classification of shapes that make up an object in advance. Furthermore, the adaptability of the system would be compromised by this process, since learning new shapes would also require these extra steps of categorizing shapes. Once a neural network is trained to a satisfactory level it may be used as an analytical tool on other data. To do this, the user no longer specifies any training runs and instead allows the network to work in forward propagation mode only.

It is also possible to over-train a neural network, which means that the network has been trained exactly to respond to only one type of input; which is much like rote memorization. If this had happened, the learning process can no longer occur.

### F. Limitations

Apart from defining the general architecture of a network and perhaps initially seeding it with a random numbers, the user has no other role than to feed it input and watch it train and await the output.

It is also possible to over-train a neural network, which means that the network has been trained exactly to respond to only one type of input; which is much like rote memorization. If this had happened, the learning process can no longer occur.

## IV. Conclusions

This paper intention was to make a brief presentation of popular image recognition methods and algorithms used in this process.

Images can be analyzed in two ways. First method implies extracting global features from image content representation and the second and the more precise one implies local feature extraction. Global features aim to describe an image using all the pixels values for example color or texture, while local features aim to detect the interest points of the image and describe them in such a manner that they can be matched in other representations of the same scene.

In order to recognize an object we need to compute its interest points and descriptors which will be compare with a collection of images. We can train an artificial neural network to recognize the specific object based on specific patterns in its aspect. In order to make the network work we must feed it with a big set of representations of the same object in order to learn the objects aspects.

## References

[1] Moataz El Ayadi, Mohamed S. Kamel, Fakhri Karray : “Survey on speech emotion recognition: Features, classification schemes, and databases”
[2] validare în vruce: https://www.openml.org/a/estimation-procedures/1
[3] Chris Harris , Mike Stephens: “A COMBINED CORNER AND EDGE DETECTOR”
[4] Lowe, D.G. : “Distinctive image features from scale-invariant keypoints” Int. J. Comput. Vis. 60(2), 91–110 (2004)
[5] Bay, H., Ess, A., Tuytelaars, T., Gool, L.: “Speeded-up robust features (SURF)”. Comput. Vis. Image Underst. 110(3), 346–359 (2008)
[6] Gil Levi :“Binary descriptors” www.gilscvblog.com
[7] Alexandre Alahi, Raphael Ortiz, Pierre Vandergheynst: “FREAK Fast Retina Keypoint”
[8] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua: “BRIEF: Binary Robust Independent Elementary Features”
[9] Ethan Rublee, Vincent Rabaud, Kurt Konolige, Gary Bradski: “ORB: an efficient alternative to SIFT or SURF”
[10] Jelmer de Vries: “Object Recognition:A Shape-Based Approach using Artificial Neural Networks”