NLP based Voice Modulation using Mel Frequency Cepstral Coefficient (MFCC) and Convolution Network

Harsh Joshi[1] Harshita Agrawal[2] Ashish Chaturvedi[3] Bharat Kumar Singh[4] Dr. Sandeep Sharma[5]
[1,2,3,4,5]Department of Electronics and Communication Engineering
[1,2,3,4,5]Dehradun Institute of Technology

Abstract— Speech processing is a challenging task, owing to the stochastic nature and high dimensional nature of dataset at hand. In this paper we propose a novel technique for extracting speech characteristics and mapping the differences in them between different speakers to generate a modulation vector. Using Speech to text conversion, natural language processing for phoneme based segmentation, and the modulation vector as suggested earlier, we aim to modulate speech characteristics of host speaker to deliver voice content (audio signal) in target speaker's voice.

Keywords: Natural Language Processing, Audio Fingerprinting, MFCC, Perceptual Hash, Deep Learning, Speech to text conversion

I. INTRODUCTION

A. NLP:

Natural language processing (NLP) is a field of computer science, artificial intelligence, and computational linguistics concerned with the interactions between computers and human (natural) languages. As such, NLP is related to the area of human–computer interaction. Many challenges in NLP involve natural language understanding, that is, enabling computers to derive meaning from human or natural language input, and others involve natural language generation. Modern NLP algorithms are based on machine learning, especially statistical machine learning. The paradigm of machine learning is different from that of most prior attempts at language processing. Prior implementations of language-processing tasks typically involved the direct hand coding of large sets of rules. The machine-learning paradigm calls instead for using general learning algorithms — often, although not always, grounded in statistical inference — to automatically learn such rules through the analysis of large corpora of typical real-world examples. A corpus (plural, "corpora") is a set of documents (or sometimes, individual sentences) that have been hand-annotated with the correct values to be learned.

Many different classes of machine learning algorithms have been applied to NLP tasks. These algorithms take as input a large set of "features" that are generated from the input data. Some of the earliest-used algorithms, such as decision trees, produced systems of hard if-then rules similar to the systems of hand-written rules that were then common. Increasingly, however, research has focused on statistical models, which make soft, probabilistic decisions based on attaching real-valued weights to each input feature. Such models have the advantage that they can express the relative certainty of many different possible answers rather than only one, producing more reliable results when such a model is included as a component of a larger system.

B. MFCC:

The first step in any automatic speech recognition system is to extract features i.e. identify the components of the audio signal that are good for identifying the linguistic content and discarding all the other stuff which carries information like background noise, emotion etc.

The main point to understand about speech is that the sounds generated by a human are filtered by the shape of the vocal tract including tongue, teeth etc. This shape determines what sound comes out. If we can determine the shape accurately, this should give us an accurate representation of the phoneme being produced. The shape of the vocal tract manifests itself in the envelope of the short time power spectrum, and the job of MFCCs is to accurately represent this envelope. Mel Frequency Cepstral Coefficients (MFCCs) are a feature widely used in automatic speech and speaker recognition. They were introduced by Davis and Mermelstein in the 1980's, and have been state-of-the-art ever since. Prior to the introduction of MFCCs, Linear Prediction Coefficients (LPCs) and Linear Prediction Cepstral Coefficients (LPCCs)

C. Phonemes

A phoneme is a basic unit of a language's phonology, which is combined with other phonemes to form meaningful units, morphemes. The phoneme can be described as "The smallest contrastive linguistic unit which may bring about a change of meaning".[1] In this way the difference in meaning between the English words kill and kiss is a result of the exchange of the phoneme for the phoneme. Two words that differ in meaning through a contrast of a single phoneme form a minimal pair.

Within linguistics there are differing views as to exactly what phonemes are and how a given language should be analyzed in phonemic (or phonematic) terms. However, a phoneme is generally regarded as an abstraction of a set (or equivalence class) of speech sounds (phones) which are perceived as equivalent to each other in a given language. For example, in English, the "k" sounds in the words kit and skill are not identical (as described below), but they are distributional variants of a single phoneme /k/. Different speech sounds that are realizations of the same phoneme are known as allophones.

Allophonic variation may be conditioned, in which case a certain phoneme is realized as a certain allophone in particular phonological environments, or it may be free in which case it may vary randomly. In this way, phonemes are often considered to constitute an abstract underlying representation for segments of words, while speech sounds make up the corresponding phonetic realization, or surface form.

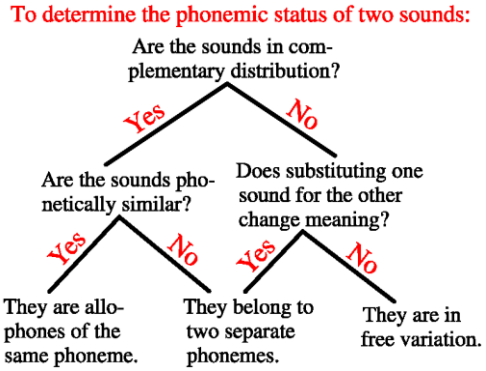

Fig. 1.1: Phonemic status determination

D. Deep Learning:

Deep learning (deep machine learning, or deep structured learning, or hierarchical learning, or sometimes DL) is a branch of machine learning based on a set of algorithms that attempt to model high-level abstractions in data by using model architectures, with complex structures or otherwise, composed of multiple non-linear transformations. Deep learning is part of a broader family of machine learning methods based on learning representations of data. An observation (e.g., an image) can be represented in many ways such as a vector of intensity values per pixel, or in a more abstract way as a set of edges, regions of particular shape, etc. Some representations make it easier to learn tasks (e.g., face recognition or facial expression recognition) from examples. One of the promises of deep learning is replacing handcrafted features with efficient algorithms for unsupervised or semi-supervised feature learning and hierarchical feature extraction.

Research in this area attempts to make better representations and create models to learn these representations from large-scale unlabelled data. Some of the representations are inspired by advances in neuroscience and are loosely based on interpretation of information processing and communication patterns in a nervous system, such as neural coding which attempts to define a relationship between the stimulus and the neuronal responses and the relationship among the electrical activity of the neurons in the brain. Various deep learning architectures such as deep neural networks, convolutional deep neural networks, deep belief networks and recurrent neural networks have been applied to fields like computer vision, automatic speech recognition, natural language processing, audio recognition and bioinformatics where they have been shown to produce state-of-the-art results on various tasks. Alternatively, deep learning has been characterized as a buzzword, or a rebranding of neural networks.

II. ANALYSIS OF CONTEMPORARY MODELS AND DATASET

A. Shazam [9]:

There is a service called Shazam, which take a short sample of music, and identifies the song. There are couple ways to use it, but one of the more convenient is to install their free app onto an iPhone. Just hit the "tag now" button, hold the phone's mic up to a speaker and it will usually identify the song and provide artist information, as well as a link to purchase the album. What is so remarkable about the service, is that it works on very obscure songs and will do so even with extraneous background noise. We can consider of any piece of music as a time-frequency graph called a spectrogram. On one axis is time, on another is frequency, and on the 3rd is intensity. Each point on the graph represents the intensity of a given frequency at a specific point in time. Assuming time is on the x-axis and frequency is on the y-axis, a horizontal line would represent a continuous pure tone and a vertical line would represent an instantaneous burst of white noise. The Shazam algorithm fingerprints a song by generating this 3d graph, and identifying frequencies of "peak intensity." For each of these peak points it keeps track of the frequency and the amount of time from the beginning of the track. Based on the paper's examples, I'm guessing they find about 3 of these points per second.

Shazam builds their fingerprint catalog out as a hash table, where the key is the frequency. When Shazam receives a fingerprint like the one above, it uses the first key (in this case 823.44), and it searches for all matching songs. They do not just mark a single point in the spectrogram, rather they mark a pair of points: the "peak intensity" plus a second "anchor point". So their key is not just a single frequency, it is a hash of the frequencies of both points. This leads to less hash collisions which in turn speeds up catalog searching by several orders of magnitude by allowing them to take greater advantage of the table's constant (O(1)) look-up time. If a specific song is hit multiple times (based on examples in the paper we think it needs about 1 frequency hit per second), it then checks to see if these frequencies correspond in time. They actually have a clever way of doing this They create a 2d plot of frequency hits, on one axis is the time from the beginning of the track those frequencies appear in the song, on the other axis is the time those frequencies appear in the sample. If there is a temporal relation between the sets of points, then the points will align along a diagonal. They use another signal processing method to find this line, and if it exists with some certainty, then they label the song a match.

B. CMU Dictionary

The Carnegie Mellon University Pronouncing Dictionary is an open-source machine-readable pronunciation dictionary for North American English that contains over 134,000 words and their pronunciations. CMUdict is being actively maintained and expanded. They are open to suggestions, corrections and other input. Its entries are particularly useful for speech recognition and synthesis, as it has mappings from words to their pronunciations in the ARPAbet phoneme set, a standard for English pronunciation. The current phoneme set contains 39 phonemes, vowels carry a lexical stress marker:

- 0 — No stress
- 1 — Primary stress
- 2 — Secondary stress

Bear in mind that this is a dictionary. If a word is not in it (or was misspelled) nothing will be returned. This applies to items such as numbers. The current phoneme set has 39 phonemes, not counting varia due to lexical stress. This phoneme (or more accurately, phone) set is based on the ARPAbet symbol set developed for speech recognition uses. You can find a description of the ARPAbet on Wikipedia, as well information on how it relates to the standard IPA symbol set.

AA	odd	AA D
AE	at	AE T
AH	hut	HH AH T
AO	ought	AO T
AW	cow	K AW
AY	hide	HH AY D
B	be	B IY
CH	cheese	CH IY Z
D	dee	D IY
DH	thee	DH IY
EH	Ed	EH D
ER	hurt	HH ER T
EY	ate	EY T
F	fee	F IY
G	green	G R IY N
HH	he	HH IY
IH	it	IH T
IY	eat	IY T
JH	gee	JH IY
K	key	K IY
L	lee	L IY
M	me	M IY
N	knee	N IY
NG	ping	P IH NG
OW	oat	OW T
OY	toy	T OY
P	pee	P IY
R	read	R IY D
S	sea	S IY
SH	she	SH IY
T	tea	T IY
TH	theta	TH EY T AH
UH	hood	HH UH D
UW	two	T UW
V	vee	V IY
W	we	W IY
Y	yield	Y IY L D
Z	zee	Z IY
ZH	seizure	S IY ZH ER

Fig. 2.1: Phoneme-Example-Translation

C. PRAAT [6]

PRAAT (the Dutch word for "talk" or "speak") is a free scientific computer software package for the analysis of speech in phonetics. It was designed, and continues to be developed, by Paul Boersma and David Weenink of the University of Amsterdam. It can run on a wide range of operating systems, including various versions of UNIX, Linux, Mac and Microsoft Windows (95, 98, NT4, ME, 2000, XP, Vista, 7, 8). The program also supports speech synthesis, including articulatory synthesis.

III. NLP CLASSIFIER

Using the CMU pronunciation dictionary we create a list of 500 words which when used as a transcript for profile modelling of speaker, then cover maximum possible variation of phonemes. To do this we parse the dictionary set in our scripts and generate a lexical tree structure. Then modelling the knapsack algorithm on the sparse tree we form our list of final dictation transcript. The process is straight forward and can be used repeatedly after updating the parent dictionary.

IV. SAMPLE ISOLATION AND SEGMENTATION

The approach derives its influence from [7], however when the classifier text as described in section 3 is known then the reverse mapping of segmented phonemes saves further processing required in audio to phone text alignment, thereby making it a verbose model

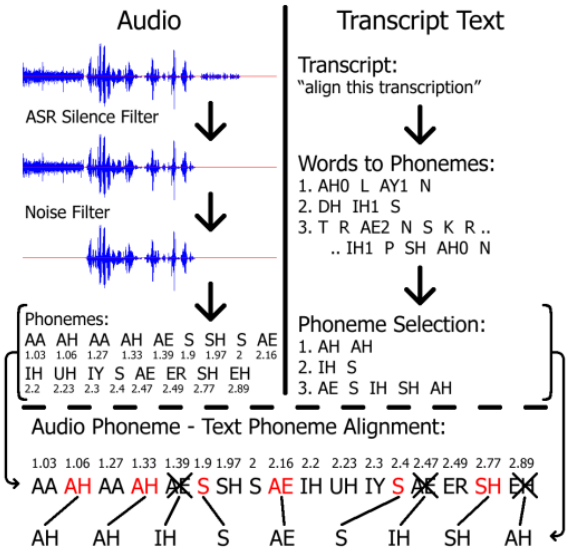

Fig. 4.1: Audio Phoneme to Phoneme Text Alignment Schematic

V. DICTIONARY PREPARATION

In this stage we map the corresponding MFCC segments of audio signal to the phoneme classification list prepared as described in Section 3 for a linear time hash preparation as the dictionary is to be indexed and modified iteratively in the section 6.

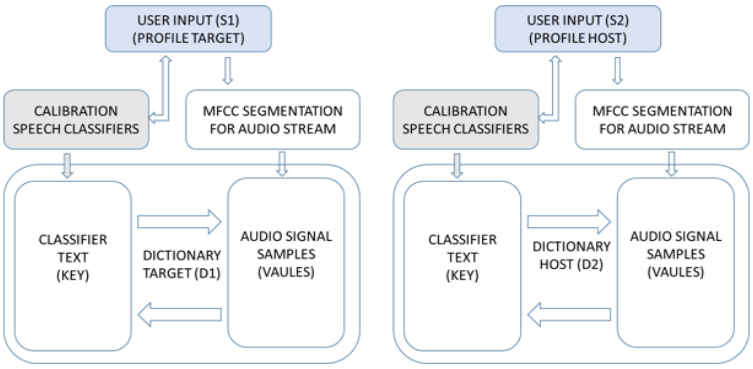

Fig. 5.1: Dictionary Schematic for the system

VI. CLASSIFIER MODULATION [5]

A. *Types of Features*

Basilar membrane in the inner ear actually analyzes the frequency content of the speech we hear. In fact, the analysis of basilar membrane can be modeled by a bank of constant Q, band pass filters. There also exist the critical bands, which give rise to the phenomenon of masking - where one strong tone or burst can mask another weaker tone within the critical band. Both MFCC and PLP capture these characteristics of our auditory system in some way; so, even though it looks strange, the same features give reasonably good performance for speech recognition, speaker recognition, language identification and even accent identification.

However, these spectral features are not very robust to noise. On the other hand, some of the time domain (temporal) features such as plosion index and maximum correlation coefficient are relatively more robust to noise

1) *Temporal Features*

The temporal features (time domain features), which are simple to extract and have easy physical interpretation, like: the energy of signal, zero crossing rate, maximum amplitude, minimum energy, etc.

2) *Spectral Features*

The spectral features (frequency based features), which are obtained by converting the time based signal into the frequency domain using the Fourier Transform, like: fundamental frequency, frequency components, spectral centroid, spectral flux, spectral density, spectral roll-off, etc. These features can be used to identify the notes, pitch, rhythm, and melody.

B. *Variation Matrix [11] [12 [13]:*

It is the most crucial stage of the system for deciding the efficiency for here we design the bank of spectral feature sets and their corresponding modulation in audio to phoneme dictionary and further modifies the dictionary for future use. The Variation matrix as shown in the Fig 7.3 comprises of these parameter with their values decided dynamically and experimentally.

VII. FEATURE MAPPING AND CLASSIFICATION [8]

So far in this system we have the required dataset to map the differences in the characteristics of both profiles. 3d Map is made defining the relation between the variation matrix parameter in section 6, phoneme, and MFCC for the corresponding profiles. The matching process in similar to edit distance matching in the DNA alignment process. However to get a good sense of approximation and avoid hard threshold decision, classification of difference in values in done on the basis of cosine similarity between the profile data sets. So effectively it boils down to a dimension for similarity index, a dimension for parameters and a dimension for time scale to describe phoneme (as the MFCC have been mapped to Phonemes in the previous sections iteratively on a many to many based mapping scheme). This information plays a crucial role in the interpolation engine of the system, and loosely derives its influence from the fingerprinting scheme used in Shazam as described in section 2.1. The following schematic (Fig 7.3) describes the process of perceptual hash preparation which represents our feature map of profiles.

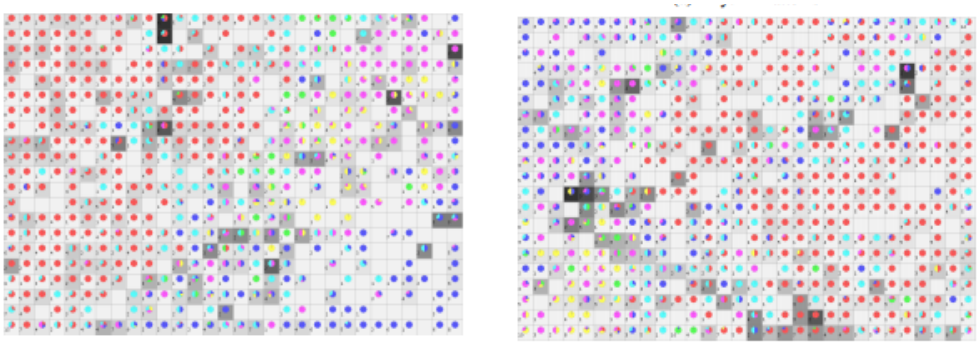

Fig. 7.1 Comparison of 2 feature sets

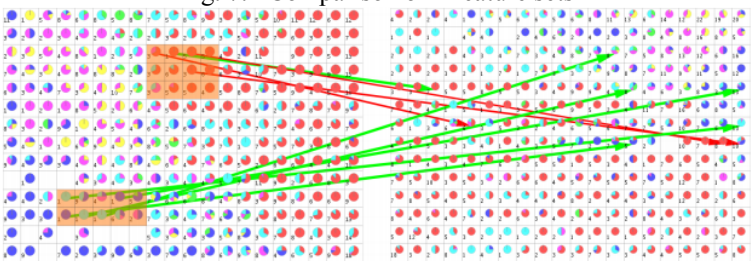

Fig. 7.2: Data Shifts Visualization

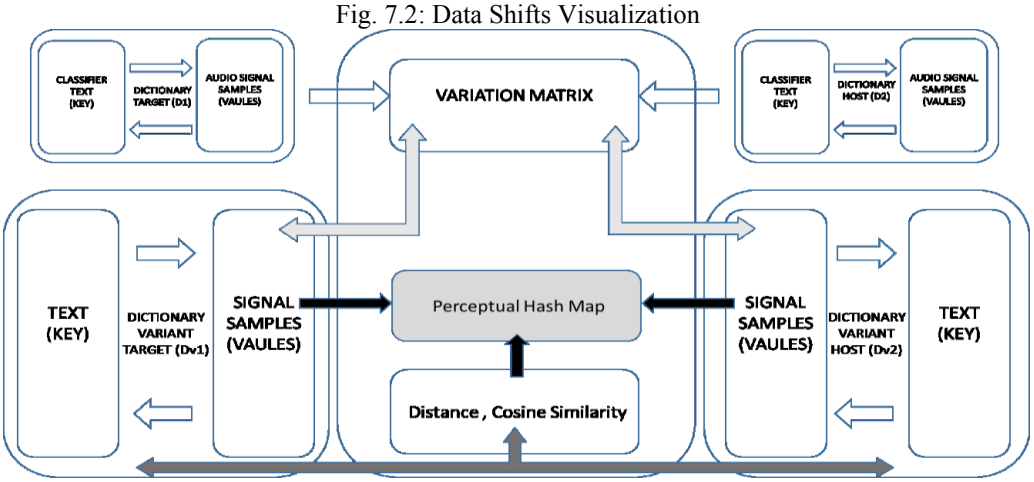

Fig. 7.3: Perceptual Hash Map Schematic for the system

VIII. CONVOLUTION NETWORK AND INTERPOLATION ENGINE [14][15][16]

It is the final stage of the system which is to be used after all the configurations are made. Interpolation engine uses the Feature map which acts as a modulation vector for varying the MFCC segmented audio stream of host to deliver speech audio stream in target profile.

The deep learning network makes use of convolutional architecture due to their merits in efficiency and performance. The convolution network used in the final stages helps in making the system more generic and less prone to the bias in accent and other local features. The modular design enables us to simply pass the perceptual map and audio data sets of the speakers into the convolution network and get the parse audio in output stream. Hardware bottleneck are overcome by the use of CUDA as offered by Theano and Torch framework. Both are compatible with the python which was primarily used for the development purposes in the prototype stage.

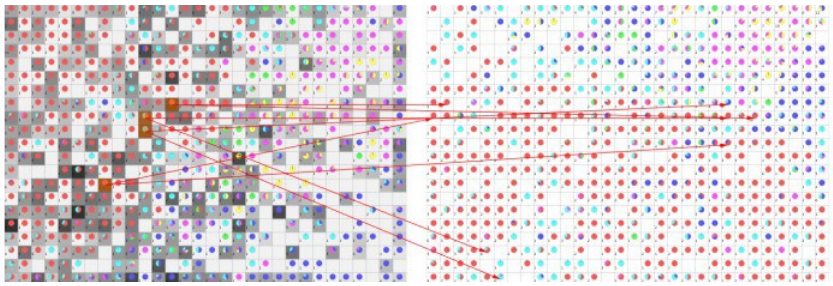

Fig 8.1: Interpolation Visualization for schematic as shown in Fig 8.2

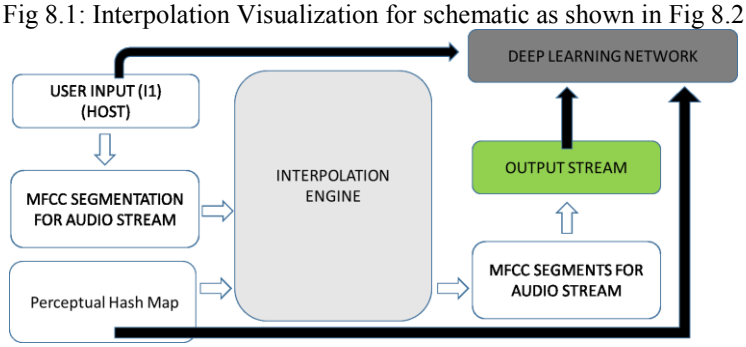

Fig. 8.2: Interpolation Engine Schematic for the system

IX. CONCLUSION

As you can see above, the final use of convolution network for deep learning overcomes the shortcoming of the initial phase of system which is heavily biased to person's accent. The limitations can be further compensated by modifying the parameters in variation matrix as described in section 6. The systems hold great importance to the speech processing systems relaying speech outputs for a custom speech profile. Many other novel and hybrid techniques used in our system can be used independently in the respective domain due to the modular nature of the system.
For more recent data, algorithm and results, you can follow the system as it evolves on the following link: https://github.com/meharshjoshi93/NLP_Modulator.

REFERENCES

[1] Davis, S. Mermelstein, P. (1980) Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences. In IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. 28 No. 4, pp. 357-366
[2] X. Huang, A. Acero, and H. Hon. Spoken Language Processing: A guide to theory, algorithm, and system development. Prentice Hall, 2001.
[3] Chomsky, N.; Halle, M. (1968), The Sound Pattern of English, Harper and Row, OCLC 317361
[4] Harris, Z. (1951), Methods in Structural Linguistics, Chicago University Press, OCLC 2232282
[5] Jakobson, R.; Fant, G.; Halle, M. (1952), *Preliminaries to Speech Analysis*, MIT
[6] Paul Boersma & David Weenink (2013): Praat: doing phonetics by computer [Computer program]. Version 5.3.51, retrieved 2 June 2013 from http://www.praat.org/
[7] Alignment of speech to highly imperfect text transcriptions. Alexander Haubold and John R. Kender
[8] Analytic Comparison of Audio Feature Sets using Self-Organising Maps. Rudolf Mayer, Jakob Frank, Andreas Rauber
[9] An Industrial-Strength Audio Search Algorithm. Avery Li-Chun Wang, Shazam Entertainment, Ltd.
[10] A Robust Feature Extraction Algorithm for Audio Fingerprinting, Jianping Chen, Tiejun Huang
[11] Ziad Al Bawab, An Analysis-by-Synthesis Approach to Vocal Tract Modeling for Robust Speech Recognition, Ph.D. Thesis, ECE Department, CMU, September, 2009.
[12] Xiang Li, Combination and Generation of Parallel Feature Streams for Improved Speech Recognition , Ph.D. Thesis, ECE Department, CMU, February 2005.
[13] Xiang Li, Combination and Generation of Parallel Feature Streams for Improved Speech Recognition , Ph.D. Thesis, ECE Department, CMU, February 2005.
[14] "Convolutional Neural Networks (LeNet) - DeepLearning 0.1 documentation". DeepLearning 0.1. LISA Lab. Retrieved 31 August 2013.
[15] LeCun, Yann. "LeNet-5, convolutional neural networks". Retrieved 16 November 2013.
[16] Ranzato, MarcAurelio; Poultney, Christopher; Chopra, Sumit; LeCun, Yann (2007). "Efficient Learning of Sparse Representations with an Energy-Based Model". Advances in Neural Information Processing Systems.