Reconhecimento de Linguagem de Sinais

# TCC de Giordano Bruno – 2014

* Reconhecimento de gestos > alternativa à periféricos
* Gestos fazem parte do pacote de formas de comunicação natural do ser humano
* Um sistema de reconhecimento de gestos auxiliaria na comunicação de deficientes auditivos
* Como está este mercado? Existem soluções comerciais?
* Duas abordagens para reconhecimento de sinais: visão computacional e luvas instrumentalizadas
* Capturar sinais – reconhece-los – transformá-los em uma saída de texto válida

Linguagens de sinais:

* Configuração da mão
* Ponto de Articulação
* Movimento
* Orientação
* Expressão facial/corporal

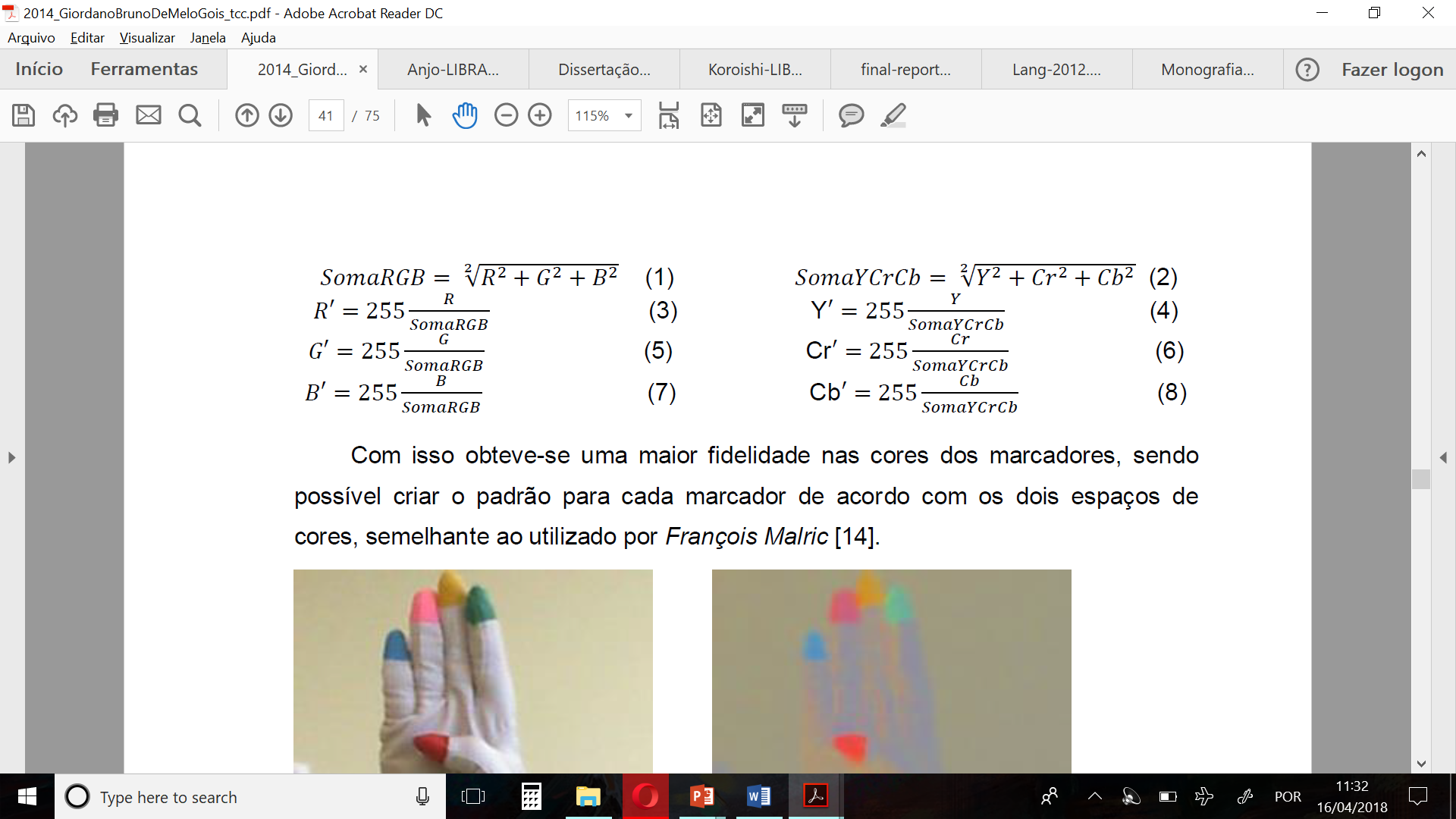
Kinect tem imagem RGB com resolução baixa: 640x480

Metodologia:

1. Escolha do alfabeto
2. Análise prévia
3. Criação de banco de imagens:
   1. Os centroides foram marcados manualmente
   2. Foi feita a contagem de imagens para uma mesma quantidade de marcadores visíveis dado cada gesto.
   3. Criação de modelos: descrição de vetores de características aceitáveis para cada letra. (pode haver mais de um)
4. Teste e validação

Sequência de passos do trabalho:

* Kinect + Câmera + marcadores coloridos.
* Kinect tem um tradeoff entre FPS e resolução: 60fps com 320x240 ou 30fps com 640x480
* A câmera fazia 30 frames/seg que não era adequado para gestos dinâmicos
* Homografia para correção do casamento da imagem da câmera com o Kinect, via OpenCV
* Calibração manual da Homografia
* Nui\_DrawSkeleton da API do Kinect identifica cabeça, pescoço, ombros, cotovelos, mãos etc.
* Kinect acha a mão, e pela homografia, sabe-se a posição desta na img rgb
* Ajuste de resolução da profundidade (eixo Z) para que o objeto mais próximo da câmera seja 255, e não o ponto imediatamente afrente da câmera. Os pontos a mais de 25 cm desse pixel mais próximo recebem o nível 0. *CSkeletalViewerApp:: Nui\_ShortToQuad\_Depth().*
* Etapa de remoção de fundo
* A resolução no plano XY não é suficiente para captar as pontas dos dedos em todos os frames. Faz-se uma média da região delimitada para a mão de 6 frames consecutivos.
* O espaço de cores RGB é o que fornece a maior diferença entre os marcadores, porém estes marcadores são susceptíveis a variações devido a iluminação. Por isso foi utilizado também o espaço de cores YCrCb. Foi feita a normalização dos dois espaços:



* Definição de valores base RGB e YCrCb para marcadores e das margens aceitáveis para cada uma das componentes.
* Busca dos pixels que são compatíveis com os critérios para encontrar os marcadores. Será marcado o centroide do blob de pixels categorizados como um marcador.
* Faz-se esta busca separadamente para os critérios RGB e para o YCrCb. A união (ou interseção) é utilizada para buscar o centroide.

Sobre a categorização:

* Para Letras com movimento foram utilizadas apenas sua configuração de mão final
* Foram classificadas a possibilidade de cada marcador aparecer em todos os gestos, podendo variar de sempre visível, pouco visível ou ocluso.
* Também foi classificada a posição da mão: frente, verso, misto, de lado.
* Critérios importantes:
  + Ordem dos marcadores
  + Quais marcadores estão visíveis
  + Distância entre eles
* Quais marcadores estão visíveis é representado binariamente: por exemplo 10110 < o que indicaria que o verde e o azul estão oclusos, as letras que tem essa combinação entrarão no grupo deste valor: 16+4+2=22. Cada combinação de dedos visíveis representará um numero diferente. Pode acontecer de uma letra ter a possibilidade de ter um dedo ocluso ou não, isto faz com que uma letra pertença a mais de um grupo.
* O método utilizado acima é utilizado para reduzir o espaço amostral para a decisão de qual letra está sendo comunicada.
* Vetor de características é dado pelas distancias relativas entre os marcadores (10 dimensões) normalizado para que não haja problema relativo a posição em Z do usuário, e 2 dimensões que representam o ângulo de rotação da mão em X e Y.
* Critério de decisão:
  + contagem de marcadores, para redução do espaço de potenciais gestos.
  + Comparação com os modelos de gestos possíveis: rotação, distancia, menor erro para desempate.

Validação

1. As entradas que geraram o modelo foram colocadas como input para verificação.
2. Entradas, cujos resultados foram péssimos devido a problemas de captura de profundidade
3. Nova base para treinamento (maior, que gerou mais classificadores) e teste feita com professor de LIBRAS.
4. Aluno refez a base, com a mesma quantidade do experimento 3.
5. Usou a base do professor e o teste do aluno e vice e versa. Resultado ruim.

Referências:

* Lamar et al, (2000): Referência para marcadores coloridos, gestos modelados por vetores de 22 dimensões
* Silva et al, (2014): ASL usando apenas informação de profundidade.
* Iterative Closest Point (ICP) é um algoritmo de alinhamento dominante na literatura que tem como objetivo recuperar uma solução de qualidade para o movimento euclidiano entre duas formas de pontos 3D.
* François Malric: dois espaço de cores

**Comentários:**

* Verificar se o SDK é compatível com este pc e/ou com o do lab
* Fazer a minha pesquisa sobre o que é o Kinect.
  + Tem versões? qual é o modelo de luciana?
  + Especificações das limitações da visão do Kinect (espaço visível, alcance, etc)
  + SDK permite python? Se não, dá pra fazer uma parte em C e outra em python?
  + 1cm de profundidade corresponde à qual variação de intensidade de pixel. É possível ajustar? Como?
* Real time analysis
  + Como fazer a passagem contínua de informação da câmera para o processamento?
* Quais são os aspectos que diferenciam os gestos? Qual deve ser a técnica utilizada?
  + Machine learning é a preferência, mas qual será o vetor de características?

# Costa Filho (2017) – A fully automatic method for recognizing hand configurations of Brazilian sign language

Objective:

* To recognize the gestures from LIBRAS (not the alphabet)
* Hand segmentation, feature extraction using LDA and PCA, and classification using novelty classifier and KNN
* Robust database was constructed: 12200 images and 200 gestures.
  + Database called: LibrasImages
  + Differente articulation point

About LIBRAS:

* 5 phonologic parameters: Hand configuration, articulation point, orientation, movement and facial expression
* 61 HC in LIBRAS

The study:

* Only depth image was used
* Steps:
  + Hand+forearm segmentation: reasoning to correct the alignment of the arm. Done with region growth with threshold and 8-connected as similarity criteria. 
  + Vertical alignment: bilinear interpolation (second Hu moments)
  + Hand segmentation: é o threshold para segmentar a mão
  + Size standardization
  + Pixel value normalization
  + Feature extraction was a bit confusing, a lot of math is exposed on the topics of PCA and LDA
  + Classification
  + Benchmark:
    - segmentation: 0.32s; feature extraction: 0.086s and classification: 0.022s.

References:

* Dong et al (2015)
  + Pixel classification from depth image (Hand x Background) using Random Forest classifier
  + Hand divided in 11 regions
  + Finger joints identification using mean-shift local mode-seeking algorithm (estimates the mass center of probability distribution of each hand region
  + Hand gesture is classified using a 13 feature joint vector as input of random forest classifier
* Silva et al (2013):
  + Accuracy of 99% :0
  + Template matching
  + Comparison metrics between two templates
  + Limitations: templates had to be obtained at the same distance and position from Kinect
* Rankun et al (2013)
  + Random Forest classifier and Generalized Learned Vector Quantization
  + 3 recognition approaches: only hand-shape data, only skeleton data, combination of those 2
  + Accuracy of 94% for the latter
* Lee et al (2016)
  + Hand-shape + hand position + movement
  + Handposition is obtained using skeleton information and decision tree
  + Hand shape is determined using PCA and SVM classifier
  + Movement direction with HMM
  + Confusion matrix to obtain probabilistic matrix to decide the word perceived.
  + Accuracy of 85%

**Comments**

* Sendo a imagem segmentada em uma forma irregular que delimita a mão, como é que se “trabalha” com esta forma irregular? (ela não é quadrada) (o ideal seria definir o que seria “trabalhar” para poder fazer esta pergunta)
* Vetor de características: se cada imagem de tamanho mxn é transformada em um vetor mnx1, temos que um conjunto de imagens é um ponto neste espaço.



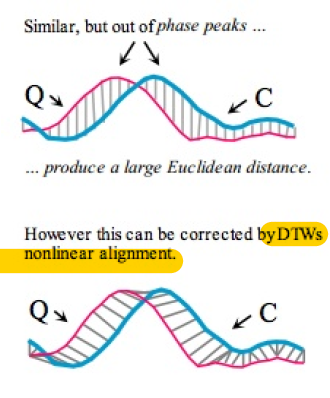
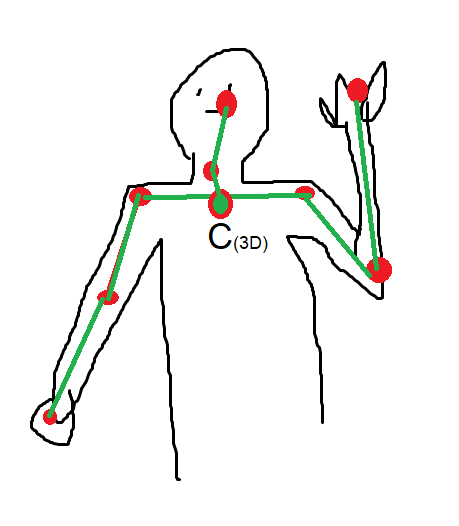






# Xu (2015) – Sign Language Translation Using Kinect And Dynamic Time Warping

* Mobile device application for gesture recognition (not mobile because of Kinect connection)
* DTW & small vocabulary (5 words only)
* Libfreenect for depth sensor input processing
* OpenNI for skeleton joint tracking and computation (randomized decision forest)
* OpenCV for feature extraction
* Implementation of a locality-constrained version of DTW with the Sakoe-Chiba-band (10 f)
* Recognition is done using a sliding window of the last N features
* Feature extraction is done with OpenNI: head, neck, shoulders, elbows and hands coordinates
* Normalization: values are not absolute locations, but position relative to C, and size normalized to D (shoulder to shoulder distance)
* Each location is concatenated into a 24-dimention vector (8\*3)



* If 90% of the gesture matches one in the database, it’s recognized.
* While receiving one gesture, DTW has to be recalculated for every database element.

References

* DTW offers difficulty for larger vocabulary because it becomes computationally expensive
* On the other hand HMM and NN are used a lot but they require a dense training phase and the recognition pipeline is lighter
* Structure sensor - mobile

**A Real-Time System to Recognize Static Gestures of Brazilian Sign Language (Libras) alphabet using Kinect – Mauro Anjo (2012)**

* Small set of gestures (A,E,I,O,U,B,C,F,L,V)
* Segmentation: Virtual Wall and Libras-specific heuristics
* Classification: MLP
* Processing rate: 63,5Hz

References

* Bretzner (2001): scale-space blob model for the hands
* Chen (2008), Fang (2007): Haar-Like features and adaBoost training
* Phu, Wysosky: MLP
* Classified by MPL to feed Hidden Markov Layer model

**Comentarios**

* Na introdução, o autor reclama que muitos dos estudos se preocupam apenas com a taxa de reconhecimento, e não fornecem informações sobre a aplicabilidade em sistemas de tempo real
* Dificuldades da segmentação:
  + Iluminação, cluttered background, roupas, oclusões, velocidade do movimento, qualidade e fps da câmera, etc
* Depth information can be obtained with stereo-vision (double camera) but it is still affected by some of the problems above and are dependent on visual properties (corners, edges, color)
* Em alguns gestos (os que a mão tocam o tronco, por exemplo) a mão não está mais afrente.

# Lee (2016) – Taiwanese

* Positions of wrist, shoulder, spine, and hip are used to localize position of hands?
* Problem divided in 3 parallel solutions:
  + Hand position
  + Trajectory
  + Hand shape recognition
* Positions of wrists are recorded as a gesture trajectory over a certain time interval
* As so, velocity, angle distance and distance between the 2 hands of gesture trajectory are extracted as features
* Extracting hand shape: wrist position for palm areas and ajustments by PCA.
* Hand palm conditions:
  + Depth variation of pixels < 20 cm
  + Pixels are no further than 70 pixels from the wrist joint
* Segmentation:
  + There may be a frame where the 2 hands are together, the separation of the palms is made by vertical projection and otsu thresholding method.
  + Palm must be aligned so it is rotation invariant (use covariance matrix and eigenvalues, PCA)
* Feature extraction:
  + Hand is divided into 5x3 nonoverlapping blocks. The percentage of hand inside each box is one of the hand features. The average depth of each box is another feature.
* Classification
  + SVM,
  + Results dropped due to low Reliability: less than 0,5 or too few palm pixels

# Makris (2015) – Hierarchical particle filtering for 3D hand tracking –

References:

|  |  |
| --- | --- |
| Bottom Up | Top Down |
| Appearance based | Model based |
| Image feature + classifier → pose (within a SET) | Parametric model of hand + search for optimal solution in parameter space |
| Big training datasets (data-driven) | Local optimization or filtering |
| no initialization required | Initialization required  Difficulty to recover from track failure |
|  | More accurate |

Hybrid Model + appearance: Appearance to guide model based optimization

* H: generative tracker (gradient descend to find optimal pose) based on color similarity combined with a part-based discriminative method. Two inicializations: previous pose and the proposed from discriminative method
* H: Random forest to form initial pose proposal while optimization scheme searches for local optimum around those.
* H: Stochastic optimization with gradient descent for fast convergence

|  |  |
| --- | --- |
| Model b.: Holistic | Model b.: Part-based |
|  | State → sub-states (parts of articulated model) |
|  | Ref: nonparametric belief propragation to approximate the posterior |

“problem with optimization based methods is that it can be trapped in local minima of cost function because they do local optimization”

Bayesian:

* Particle Filter (PF), hierarchical grid-based Bayesian filter, Partitioned Sampling (PS), Annealed Particle Filter (APF)
* Ref: PF based tracker performs PCA to obtain a 7D subspace that characterizes hand motion. The valid hand configurations lie in a Manifold, cuja hypothesis is generated by a dynamic model
* Ref: PF with stochastic meta-descent (gradient based local optimization) 26 DOF

# Pigou (2014) – SLR using CNN – Belgium

* CNN (64px-64px-32frames) 3 layers. All parameters given.
  + 2 identical CNNs, one for upper body and one for hand features
* Depth information. Noise removal of depth map.
* Input is RGB, D, zoom RGB, and zoom D
* ReLU + dropout + Nesterov’s accelerated gradient descent (NAG) + local contrast normalization
* GPU acceleration
* Experiments: Performance analysis with Incremental addition to the architecture!!
* Real time. Temporal segmentation not very explained. Sliding window

# Pigou (2016) – Sign Classification in SL Corpora with DNN – Belgium

* input image sequence consists of 8 frames of size 128x128
* inputs are rotated, shifted ans stretched randomly during training (data augmentation)
* C³32 - P - C³64 - P - C³128- P - C³256 - P - C³512 - P - D2048- D2048 - S
* attempt cross-domain feature learning (type of transfer learning) training a shared model on both corpora

# Pigou (2016) – Beyond Temporal Pooling – Belgium

# Pigou (2017) – Gesture and SLR with Temporal Residual Networks – Belgium

* temporal convolutions, residual networks, batch normalization and exponential linear units
* Approach the problem as a continuous framewise classification task, where the temporal locations of gestures and signs are NOT given during evaluation
* the model is inherently CNN, but also make use of temporal convolutions and recurrence
* Overview of I/O
  + produce predictions every frame
  + a number of frames are fed into the model, captured by a sliding window
  + outputs the prediction for the middle frame or all input frames (2 experiments)
* Preprocessing
  + RGB -> Grayscale
  + resized to 128x128
  + previous frame subtracted from current frame
* Model
  + Residual building blocks
    - Spatial convolutions
    - folowed by temporal convolutions
    - folowed by batch normalization
      * shifts internal values to a mean of zore and scale to variance of 1 in every layer. Solves the covariant shift problem.
    - ELU (better regularization)
    - repeat but before ELU input is added (how is it added?)
  + initial layer: 3D convolutional layer 7x7x7,16 and stride 1x2x2. (memory limitation)
  + 8 Residual blocks
  + Total: 17 conv layers. result: feature map
  + 3D Average, bidirectional lstm that enables processing of sequences in both temporal directions
* Training
  + backpropagation through time
  + cross-entropy loss function using mini-batch gradient descent with Adam update Rule
  + improved training convergence in comparison to SGD with nesterov momentum
  + early stopping
  + mini-batch size 24
  + learning rate of 10⁻³
  + exponential learning rate decay
  + weight initialization with random orthogonal initialization method
  + data augmentation (specified)
* PostProcessing
  + modus-filter size 39 on the final framewise predictions. Smooths noisy redictions
* Experiments
  + dataset 25fps
  + 70% Training, 20% test, 10% validation
  + Only frames of known signs are considered for evaluation
  + top-N accuracy measure
  + confusion matrix
  + used a form of transfer learning for different datasets, using the same structure, recalculated only the softmax
* Important comments:
  + Gesture data does not have silent labels. So the configuration that tries to predict every frame’s gesture could try to model too much silent annotations while the data does not have that labeled
  + be careful with class imbalance accuracy measures will be highly skewed
  + 3D CNN = 2D + 1D

# Molchanov (2015) – Hand Gesture Recognition with 3D CNN – NVIDIA

* 2x CNN
* Made sure to make every gesture sequence to 32 frames using nearest neighbor interpolation. (NNI)
* Interleave image gradient and depth frames
* Nesterov’s accelerated gradient descent (NAG), stochastic gradient descent with mini-batches. Formula given.
* Random initialization
* Weight decay
* Dropout compensation
* Online and offline data augmentation
  + Online: reverse ordering, mirroring and both together
  + Offline:
    - Spatial: affine transformations, spation elastic deformation, fixed pattern dropout and random dropout.
    - Temporal: scaling the duration of a sequence, temporally translating and elastic deformation.
* No validation set
* Test set based on leave-one-out

# Garcia (2016) – Real-time ASL recognition with CNN – Stanford

* Real-time my ass .\_. Captures one frame per second and lets the user know to move on to the next letter
* a-y fingerspelling
* webcam
* Input
  + random crops of 224x224 of the (resized) 256x256
  + zero-center the data by subtracting the mean image from ILSRVC2012
  + horizontal flips of images from dataset
* Pipeline
  + Web application platform to receive input real time indicating word start and end
  + CNN - Caffe framework.
  + Once the user has indicated they’re finished signing one word. It inputs the top-5 letter for each letter of the sequence in a language model
  + Custom unigram language model based on the Brown Corpus
* Training
  + Transfer learning using GoogleNet pre-trained on the 2012 ILSVRC dataset (1000 different object and classes). Reinitialization of 1 or 2 layes.
  + Tested effectiveness of altering a variety of the pre-trained weights at dif. depths using Xavier initialization
  + initial base learning rate of 1e-6
    - decreased by factprs ranging from 2 to 100
  + After seeing the noisy outputs they tried changing the batch size value from 4 to 20 (they trained a net using a Lighting Memory-Mapped Database)
* Softmax loss cost function should output probability while SVM loss outputs scores

# Rao (2018) – Deep CNN for SLR

* Indian Sign Language. 1 dense softmax
* CNN
  + 4Conv Layers(16, 9, 5, 5 windows), 5 ReLU, 2 stochastic pooling
  + Input 128x128x3
* Selfie mode continuous video using a mobile front camera
* Visual attention based framework (not explained but there is a reference article)
* Compared max, mean and stochastic pooling
* Equations for softmax, cost function, hypothesis function, stochastic pooling, tanh activation function (though they said they used ReLU)

# ~~Kumar (2018) - Training CNNs for 3D SLR with color texture coded joint angular Displacement maps~~

* Uses RGBD and CNN
* Joint angular displacements maps are used to interpret each sign. It encodes the sign as a color texture image
* Sounded complicated >-<

# ~~Salian(2017) – Proposed System for SLR –India~~

* Not tested!!!
* Fingerspelling?
* Talked about Background subtraction, convex hull detection and counting the defects in the resulting convex hull for pre-processing.
* Calibration: user will be asked to position their hand properly in the window

# Yasir (2017) – Bangla SLR using CNN

* Leap motion controller (LMC)
* HMM -> CNN -> output
* None of the parameter were specified. Nor the initialization, hyperparameter nor the output type.
* The latest reference is 2015, come on guys .\_.

# ElBaday (2017) – Arabic SLR with 3D CNN – Egypt

* 25 words from 2 different signers
* They say they use depth information but it sounds like depth in time, as in sequence of frames
* Comparison between frame rates: 10, 30, 50. Studying the tradeoff between missing information and having redundancy
* Frame selection according to the priority calculated from Scoring Algorithm
* Softmax, canny.
* 2 Conv Layers. Specified decay, momentum and learning rate BUT NOT activation or cost functions

# Oliveira (2017) – Irish SLR using PCA and CNN – Ireland

* Very diverse related work listing, but also a bit old (2011-2014). They list HMM, Fourier Analysis, Points of Interest, PCA, Orientation Histograms, Kalman Filter, Local Binary patterns, geometric features, support vector machine
* 23 letters Fingerspelling (images)
* Diversity score to remove redundancy in the dataset
  + feature extraction by splitting the image in a KxK grid, edge detection and histogram of the foreground pixels in each cell
* PCA:
  + showed better results if blurried gaussian (36x36), var = 60
  + The more eigenvectors used the better accuracy
  + Nice looking 3d plot and analysis of images in the extremes of each dimension
* CNN: 4 conv layers, Relu, 2 FC, softmax
  + Categorical cross emtropy as loss function
  + Adadelta optimizer with learning rate of 1
* Amazing plots and visual results!!!!!

# Pansare (2016) –Vision-Based Approach for American SLR using Edge Orientation Histogram – India

* Real-time static alphabet
* Mixed lighting conditions and complex backgounds. webcam
* Skin detector, median and gaussian filter, morphological operatios and blob extraction
* CompareHist descriptor, Sim-EOH algorithm, K-Cluster-EOH-Matching algorithm
* They do not specify how these technics are used to obtain the result

# Yang (2017) - Video-Based Chinese SLR using CNN

* Upper body images centered on hand are input into CNN
* Preprocessing
  + Face detection via Harr feature classifier
    - opencv cascade classifier supporting Harr feature
  + Detection of Hand Region via skin-color detection
    - RGB transformed to YCbCr color space
    - Cr [133,173], Cb [77,127], Y [0, 255] and R should be > than G,B
    - 2 largest contours to calculate the hand region
    - minimal area threshold set to 370
    - calculate bounding rectangle centered in the center of hand region of size ???
  + Background removal
    - compute mean image over our training and testing datasets and subtract the mean from each input image (all negative numbers are replaced by zero)
  + Color space transform
    - HSV is closer to human perception of color (hue detect uniform color regions)
    - Hue is used as single channel input!!
* I/O
  + input: Hue channel, size: ???
* Model architecture
* Training
* Experiments