

# X ENCONTRO ACADÊMICO

MODELAGEM	COMPUTACIONAL
-----------	---------------

## DESENVOLVIMENTO DE UMA FERRAMENTA PARA GERAÇÃO AUTOMÁTICA DE UM BANCO DE DADOS COM BASE EM SISTEMA MECÂNICO SIMULADO PARA TREINAMENTO DE UMA DEEP NEURAL NETWORK-DNN

**Daniel de M. L. dos Santos**

**Francisco S. M. Sinfrônio**

**André B. Cavalcante**

**Allan K. Barros**

dantmec@gmail.com

kjvida@mac.com

abcborges@gmail.com

akduailibe@gmail.com

UFMA - Universidade federal do Maranhão

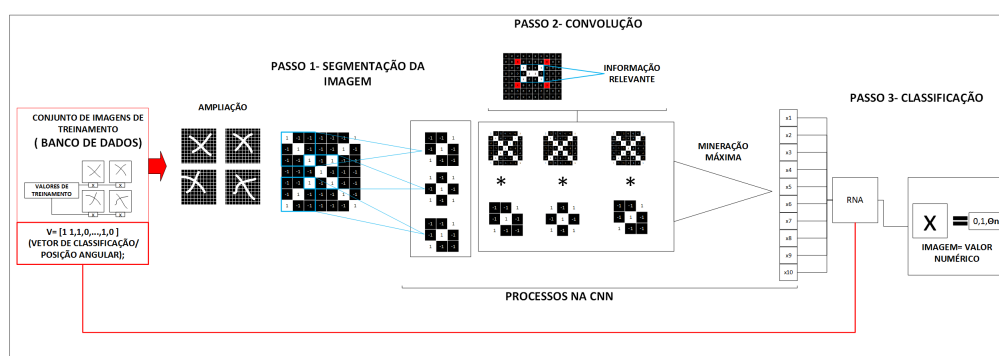
Av. dos Portugueses, 1966 - Bacanga, 65085-580, MA, São Luís, Brasil

**Resumo.** *As Redes Neurais Profundas ou DNN (do inglês Deep Neural Networks) tem sido utilizadas como instrumento de extração de padrões, extração de características de imagens e classificação sem a necessidade de processamento prévio. Para o caso de Redes Neurais Convolucionais com aprendizado supervisionado, para finalidade de classificação, são necessários portanto a geração, tanto de um banco de dados prévio com milhares de imagens para o aprendizado da rede, como um vetor de valores desejados (valores de ângulo no presente trabalho), requerendo implementação de sensores e adaptações em sistemas mecânicos reais. Objetivo do presente estudo é a criação automática de um banco de dados de treinamento com base em uma maquete virtual, onde as variáveis angulares e variáveis de posicionamento são geradas com base em um modelo matemático, previamente desenvolvido. Por fim é realizada uma técnica conhecida como Chroma-Key, onde, para cada imagem, ocorre uma mudança de ambiente ou cenário onde estará inserido o sistema mecânico virtual.*

**Palavras-chave:** *Redes Neurais Profundas, Banco de Dados, Automatização, Modelo Virtual, Imagem.*

# 1 INTRODUÇÃO

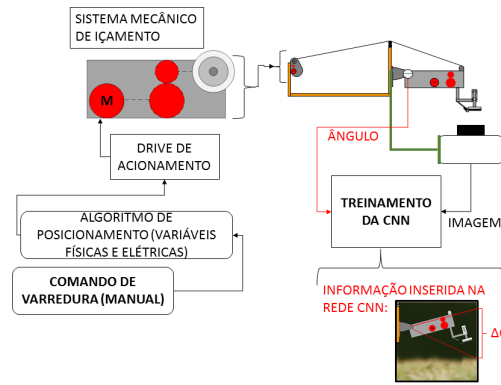
A extração de características de alta ordem para classificação de imagens, por redes neurais, sem a necessidade de processamento prévio, tem sido desenvolvida com a finalidade de aplicar uma ferramenta menos limitada à problemas considerados complexos (problemas onde ocorrem translações ou rotações do objeto a ser classificado na imagem, por exemplo). Redes Neurais Convolucionais (classe específica de Redes Neurais Profundas-DNN, voltada a processamento de imagens) são utilizadas com sucesso para classificação supervisionada e não-supervisionada de imagens como no trabalho de Xie et al. (2016). As redes Neurais Convolucionais foram concebidas posteriormente à análise dos resultados de Hubel. (1981) sobre a forma como o córtex visual de animais respondia a estímulos com orientação espacial variável. O processo de extração de característica(s) de um conjunto de imagens por uma Rede Neural Convolucional, pode ser visualizado na figura 1:



**Figura 1: Modelo de Treinamento de uma Rede Neural Convolucional. Fonte: Autor.**

A Rede Neural Convolucional segmenta a imagem (em escala de cinza ou colorida) através da convolução de uma pequena parte da matriz de pixels sobre toda a imagem, formando matrizes de menor ordem, convertendo-as em submatrizes (através da média dos valores convoluídos, por exemplo, dependendo da função de agrupamento). A subamostragem da imagem da camada anterior gera um mapa de características redundantes. Após esta etapa, características de alta ordem são agrupadas em um vetor de valores numéricos, os quais, serão tratados por uma estrutura usual de combinador linear com uma respectiva função de ativação, robusta ao problema de classificação (para o problema em estudo, a função softmax é utilizada como uma probabilidade normalizada).

Há necessidade portanto, de um vasto banco de dados para o treinamento das Redes Neurais Convolucionais (CNN), no caso em estudo, formado por aproximadamente 10.000 imagens associadas a valores de ângulo em diferentes ambientes. A complexidade da imagem ou o tamanho da mesma está diretamente relacionada ao número de camadas convolucionais necessárias, assim como ao número de dados de imagem para o treinamento (Matthew et al., 2013). Na figura 2 é observado o processo de aprendizado de posicionamento pela CNN no sistema mecânico em estudo:



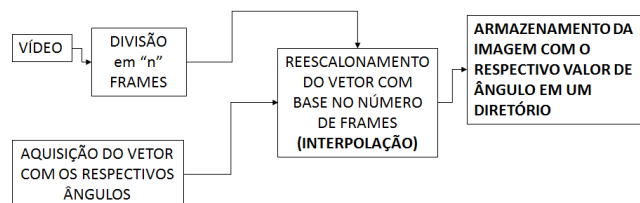
**Figura 2: Modo de Treinamento da Rede Neural Convolutacional, a ser implementado no sistema de posicionamento. Fonte: Autor.**

A representação da posição angular (imagem vista no canto direito inferior da figura 2) e seu respectivo  $\Delta_\theta$ , necessário ao posicionamento desejado, serão inseridos na entrada da CNN onde deverão ser extraídas características de inclinação e posteriormente comparação ao rótulo ( $\Delta_\theta$ ) da respectiva imagem. O cálculo do ( $\Delta_\theta$ ) é realizado através da medição feita pelo sensor de ângulo visto na figura 2, além de parâmetros anteriormente inicializados (ex: considera-se o menor valor de ângulo medido pelo sensor no processo de treinamento como o ângulo desejado para o qual a sonda deverá ser posicionada).

## 2 DESCRIÇÃO DA FERRAMENTA-MÉTODO

O modelo da ferramenta tem como base de funcionamento a extração inicial de quadros ou imagens do vídeo (sensor de imagem com tecnologia *Complementary Metal Oxide System-CMOS*), com resolução aproximada de dois Mega Pixels- 2MP), juntamente com aquisição dos valores de ângulo, medidos por um textithardware de aquisição de dados baseado na plataforma Arduino. Estes valores são posteriormente transferidos pela porta serial à um servidor (taxa de transferência de dados de cento e quinze mil e duzentos bits por segundo- 115.200 baud), digitalizados por um conversor analógico- digital com resolução de dez bits. A medida em que os valores são adquiridos ocorre sua concatenação e posterior reescalonamento com base no número de quadros (imagens).

A estrutura do algoritmo para geração automática do banco de dados, será mostrada na figura 3:



**Figura 3: Ferramenta proposta para automatização do banco de dados de treinamento. Fonte: Autor.**

A ferramenta (observada no fluxograma da figura 3) necessita de um vídeo em formato *Audio Video Interleave- AVI* do sistema de posicionamento em funcionamento (no presente trabalho foi gerado um vídeo com a operação de içamento), associado com os valores adquiridos pelo sensor de ângulo implementado. O modelo proposto acima tem como finalidade reduzir o tempo de implementação de uma rede neural convolucional em sistemas físicos reais, haja vista que o banco de dados será criado automaticamente. O reescalonamento do vetor é necessário para que haja bijeção entre o quadro (imagem) e o vetor de ângulo (mecanismo utilizado-interpolação).

## 2.1 Aquisição de sinais no modelo real

A aquisição do sinal de ângulo foi realizada através de um sensor potenciométrico com curva linear de resistência e um sensor de imagem CMOS, já citado anteriormente, cujos parâmetros estão organizados na Tabela 1:

**Tabela 1: Características principais do sensor potenciométrico utilizado na medição de ângulo no hardware de aquisição de sinais**

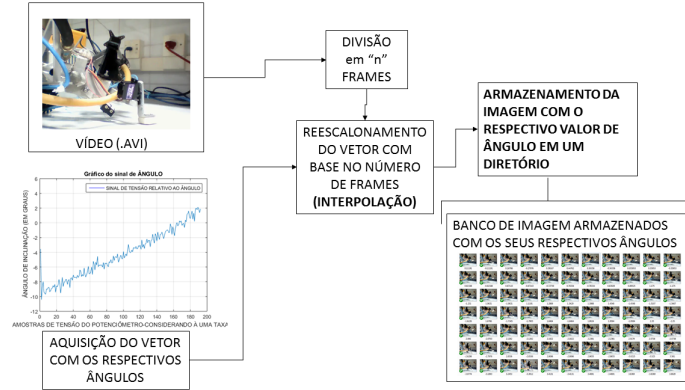
Característica	Unidade	Valor
Constante Angular	$K (Ohm/rad)$	2.3873e+03
Valor de Resistência	$R (Ohms)$	10e+03
Delta de Varredura	$\theta_{max} (rad)$	4.1888
Material	Resistor de trilha de carbono	

O sensor de imagem, implementado na parte lateral direita da plataforma robótica (orientação da câmera voltada para o sistema de sondagem, como visto na figura 2), possui as características citadas na Tabela 2:

**Tabela 2: Características principais do sensor de imagem utilizado**

Característica	Unidade	Valor
Taxa de Quadros (utilizada no projeto)	$(FPS)$	25
Resolução	1600x1200 (pixels)	1920000.00

O vídeo adquirido pelo sensor CMOS, assim como o sinal do sensor potenciométrico descrito na tabela acima, foram extraídos sem nenhuma filtragem prévia, tentando manter o sinal mais próximo possível dos valores adquiridos em cada frame. Aplicados à ferramenta proposta, geraram um banco de dados com frames associados a valores de ângulo de inclinação. Esse processo de aquisição e geração automática do banco de dados no modelo real será apresentada na figura 4:



**Figura 4: Geração do banco de dados do sistema real de sondagem pela ferramenta proposta. Fonte: Autor.**

A aquisição do valor de inclinação no sistema de posicionamento real, absorve as imperfeições de construção e instalação do sensor potenciométrico utilizado (sensor de uso comercial), tornando ruidoso o sinal de ângulo de inclinação adquirido (sinal ruidoso visto graficamente no canto inferior esquerdo da figura 4). Houve então, a necessidade da montagem de um modelo animado de maior simplicidade quanto à forma e cenário de operação, com movimentos correspondentes a um modelo matemático previamente desenvolvido.

## 2.2 Método de interpolação

O reescalonamento do vetor com os valores de ângulo exigiu uma matriz quadrada, cuja primeira linha fosse formada por todo o vetor de ângulos medidos (tanto no modelo real, como no modelo virtual animado). Considerando  $n_\theta$  como sendo o número de elementos do vetor  $\vec{v}_\theta$  (vetor de posição angular que formará a primeira linha da matriz  $M_\theta$ ), é necessária uma matriz para o reescalonamento, inicialmente formada em sua primeira linha pelo vetor  $\vec{v}_\theta$ , tendo as outras  $n_\theta - 1$  linhas preenchidas por elementos nulos e,  $n_\theta$  colunas. A solução inicialmente adotada no projeto para a matriz quadrada  $M_\theta$  de ordem  $n_\theta$  está apresentada na "Eq. (1)":

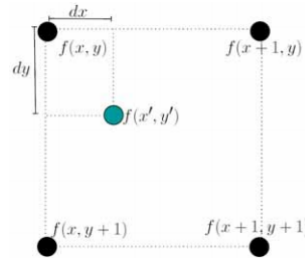
$$M_\theta = \begin{pmatrix} \theta_1 & \theta_2 & \dots & \theta_n \\ 0 & 0 & \dots & 0 \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ 0 & 0 & \dots & 0 \end{pmatrix} \quad (1)$$

Matriz  $M_\theta$

A partir do número de quadros extraídos do vídeo  $n_{frames}$  e da ordem da matriz quadrada  $M_\theta$ , o fator de escala  $R_e$  para nova matriz está inferido na "Eq. (2)":

$$R_e = \frac{n_{frames}}{n_\theta} \quad (2)$$

O método de interpolação utilizado no presente estudo é do tipo bilinear. Em geral esse método é utilizado em aplicações voltadas à reamostragem de imagens, com finalidade de minimização do efeito de serrilhamento ocasionado pelo método de interpolação do vizinho mais próximo ou "Nearest Neighbor" (Dourado.,2014). O método adotado realiza interpolações lineares horizontais e verticais entre os elementos da matriz de reescalonamento. Considerando uma imagem como uma  $f(x, y)$  na figura 5, será representada graficamente a interpolação bilinear em uma matriz:



**Figura 5:** Esquema da interpolação bilinear aplicado à uma função  $f(x,y)$ . Fonte:(Dourado.,2014).

A equação que demonstra a interpolação bilinear:

$$f(x', y') = (1-d_x)(1-d_y)f(x, y) + dx(1-d_y)f(x+1, y) + (1-d_x)d_yf(x, y+1) + dx d_y f(x+1, y+1) \quad (3)$$

Pode ser observado na "Equação (3)" um novo ponto  $(x', y')$  modelado pelos valores das linhas e colunas adjacentes, ponderado pelas distâncias ao redor do valor desconhecido. O método possui como característica a redução dos valores interpolados, tanto entre os valores das colunas (devido à interpolação com os elementos nulos da matriz  $M_\theta$ ), como entre os valores das linhas (a interpolação horizontal também é ponderada pelas distâncias verticais).

## 2.3 Chroma-Key

O Chroma-Key é uma técnica de edição de imagem desenvolvida com a finalidade de modificar o cenário ou ambiente virtual em que um objeto qualquer possa estar inserido, de acordo com critérios de cor previamente estabelecidos (Aprile et al., 2007). O sistema de cor das imagens inseridas na ferramenta é o sistema RGB (*Red, Green, Blue*- fazendo referência as matrizes ou planos de cor, contidos em cada imagem ou quadro adquirido). Para realizar a mudança de cenário, ou mudança de fundo, foram detectados os limiares de cor em cada plano. No caso em estudo, foi utilizado um fundo branco em um modelo virtual desenvolvido, desta forma, a parte do algoritmo que compõe a inicialização do limiar (responsável pela detecção de fundo) é um vetor  $\vec{v}_{\text{limiar}}$ , cuja forma é demonstrada na "Eq. (4)":

$$\vec{v}_{\text{limiar}} = [\text{limiar}_{\text{red}}, \text{limiar}_{\text{green}}, \text{limiar}_{\text{blue}}] \quad (4)$$

Os valores de limiar dos pixels  $[\text{limiar}_{\text{red}}, \text{ }]$ ,  $[\text{limiar}_{\text{green}}, \text{ }]$  e  $[\text{limiar}_{\text{blue}}, \text{ }]$  são inicializados com os respectivos valores de  $[200, 200, 200]$ , tendo como padrão para intensidade de cor o valor máximo de 255 (a cor "branca" ou de intercessão de espectro em uma imagem é

formada por um vetor  $\vec{v}$  com intensidades [255, 255, 255]). A forma de extração do objeto no fundo branco é mostrada na matriz formada pela seguinte função lógica:

$$M_{(red/green/blue)} = \begin{cases} 1 & \text{if } (\vec{\text{pixel}}[\mathbf{R}, \mathbf{G}, \mathbf{B}] - \mathbf{v}_{\text{limiar}}) \leq [55, 55, 55] \\ 0 & \text{if } (\vec{\text{pixel}}[\mathbf{R}, \mathbf{G}, \mathbf{B}] - \mathbf{v}_{\text{limiar}}) > [55, 55, 55] \end{cases} \quad (5)$$

A "Equação (5)" extrai a máscara onde será adicionado o cenário, sendo  $\vec{\text{pixel}}[\mathbf{R}, \mathbf{G}, \mathbf{B}]$  o valor do respectivo pixel (considerando um vetor) da imagem original em fundo branco varrido pela função lógica. Desta forma, para montagem do cenário ao fundo do objeto, recorre-se a equação "Eq. (6)":

$$I_{final(i,j)} = \text{pixel}_{red(i,j)} M_{red(i,j)} + \text{pixel}_{green(i,j)} M_{green(i,j)} + \text{pixel}_{blue(i,j)} M_{blue(i,j)} \quad (6)$$

Sendo  $M_{red(i,j)}$ ,  $M_{green(i,j)}$  e  $M_{blue(i,j)}$  os respectivos elementos da matrizes  $M_{red}$ ,  $M_{green}$  e  $M_{blue}$  resultantes da matrizes geradas pela função lógica  $M_{(red/green/blue)}$  referente à "Eq. (5)" e,  $\text{pixel}_{red(i,j)}$ ,  $\text{pixel}_{green(i,j)}$ ,  $\text{pixel}_{blue(i,j)}$  os respectivos pixels do cenário de fundo inseridos em cada plano de cor. Desta forma a imagem ficará conforme apresentado na figura 6:



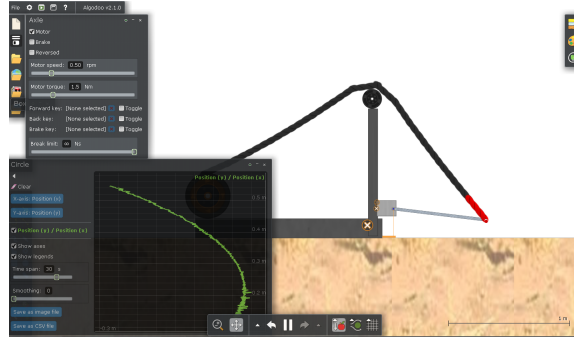
**Figura 6: Demonstração das etapas do kroma-key aplicado a cada quadro. Fonte: Autor.**

Pode ser visto no canto superior esquerdo da figura 6, o modelo virtual (cenário de fundo em cor "branca"), o cenário de fundo onde o objeto ou modelo será inserido (canto superior direito) e posteriormente a imagem ou quadro da sonda montada no cenário desejado (centro).

### 3 MODELO VIRTUAL PROPOSTO PARA COMPARAÇÃO

O modelo virtual simplificado foi desenvolvido no software *Algodoo*<sup>®</sup> (*software* livre de engenharia física, criado por Emil Ernerfeldt, nomeado inicialmente como "Phun"), liberado em 17 de Dezembro de 2007, sofrendo atualizações desde então. O modelo criado neste software leva em consideração valores reais da plataforma como massa, gravidade, resistência do ar, atrito, proximidade com os pontos de mudança de velocidade do mecanismo (caracterizados

tecnicamente como "Pontos Mortos" do mecanismo), tornando-o o mais próximo da realidade. As posições da extremidade da sonda são armazenadas em um arquivo ".csv", onde posteriormente serão extraídas pela ferramenta proposta e transformadas em valores angulares. O modelo virtual em simulação no software será apresentado na figura 7:



**Figura 7: Modelo virtual animado em plataforma virtual de desenvolvimento. Fonte: Autor.**

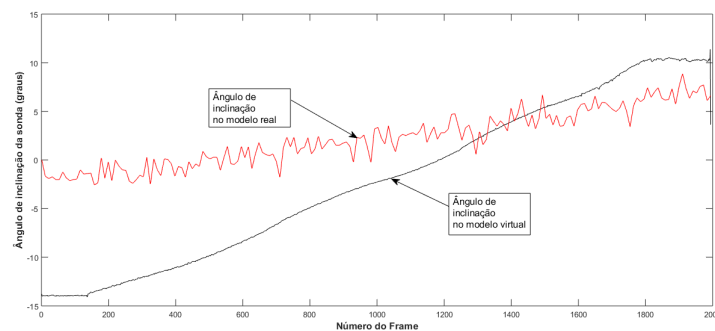
A equação que transforma os valores de posição do ponto de extremidade (circunferência vermelha na extremidade da sonda, vista na figura 7) é representada por:

$$\theta_{cc} = \sin^{-1}\left(\frac{y_{cc}}{\sqrt{x_{cc}^2 + y_{cc}^2}}\right) \quad (7)$$

Sendo  $\theta_{cc}$  o ângulo resultante calculado através das posições corrigidas  $x_{cc}$ ,  $y_{cc}$  em relação aos pontos centrais da circunferência (deslocada no ponto de rótula), formada pelo sistema de posicionamento.

## 4 RESULTADOS ENCONTRADOS

Foram realizadas aquisições finais quanto aos valores de ângulo, em uma operação de içamento, no sistema de posicionamento real, assim como no modelo virtual. O gráfico dos vetores reescalados  $\vec{v}$  são apresentados na figura 8:



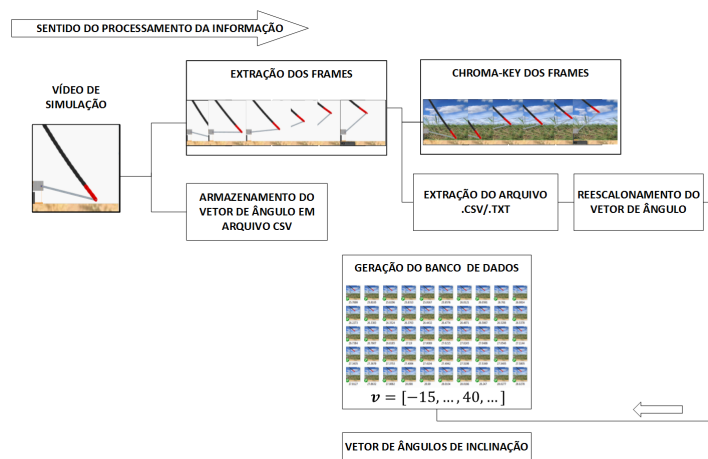
**Figura 8: Gráfico de comparação entre o vetor de aprendizado gerado no teste real e o vetor de aprendizado gerado no modelo virtual. Fonte: Autor.**



Pode ser observado no vetor de posição angular (adquirido pelo sensor no modelo real), as imperfeições construtivas e interferências na aquisição gerando um sinal ruidoso (linha vermelha no gráfico da figura 8), em comparação ao sinal adquirido no modelo virtual (linha preta no gráfico). Há também um aumento da faixa de deslocamento angular que, no modelo virtual, consegue alcançar um  $\Delta_\theta$  de aproximadamente  $25^\circ$ , em comparação à um  $\Delta_\theta$  de aproximadamente  $7^\circ$  no modelo real.

## 4.1 Contribuição Científica da Ferramenta

A ferramenta desenvolvida torna o uso de Redes Neurais Convolucionais mais robusto à problemas de engenharia e visão computacional. Isso ocorre através da geração automática do banco de dados de treinamento, onde seria necessária uma exaustiva aquisição de dados particular a cada problema. Na figura 9 a ferramenta está sendo aplicada em um modelo virtual:



**Figura 9: Geração do banco de dados do modelo virtual em cenário desejado. Fonte: Autor.**

Pode ser visto na figura 9 a geração automática do banco de dados no referido modelo computacional, podendo ser replicado em vários outros cenários (minimizando a possibilidade de overfitting na rede neural), reduzindo assim custos de implementação, ou melhorando o aprendizado de máquina em ambientes totalmente simulados.

## 4.2 Permissão

O autor Daniel de Matos Luna dos Santos é responsável pelas informações publicadas neste artigo.

## AGRADECIMENTOS

A Deus, ao Programa de Pós - Graduação em Engenharia Elétrica e ao Laboratório de Processamento da informação Biológica (PiB), que forneceram subsídios físicos e conhecimento na realização deste trabalho.

## REFERÊNCIAS

Xie, J., L., Yang, Z.C.Song., & Wying, Hu., 2016. A Theory of Generative ConvNet- *Department of Statistics, University of California, Los Angeles, CA, USA.*

Hubel, H.,D., 1981. EVOLUTION OF IDEAS ON THE PRIMARY VISUAL CORTEX. *Harvard Medical School, Department of Neurobiology, Boston, Massachusetts, U.S.A.-Nobel lecture, 8 December 1981.*

Matthew,D.,Zeiler,& Fergus, R., 2013. Visualizing and Understanding Convolutional Networks- *New York University.*

Dourado,.W.B., 2014. Avaliação de Técnicas de Interpolação de Imagens Digitais. *Universidade Estadual Paulista. Faculdade de Ciências e Tecnologia de Presidente Prudente Programa de Pós-Graduação em Matemática Aplicada e Computacional*

Aprile,M., A. Colombari, A. Fusiello, V. Murino., 2007. SEGMENTATION AND TRACKING OF MULTIPLE OBJECTS IN VIDEO SEQUENCES. *Dipartimento di Informatica, University of Verona.*