

Universidade Federal do Rio de Janeiro

Escola Politécnica

Departamento de Eletrônica e de Computação

**Identificação automática de alvos padronizados através de
redes neurais**

Autor:

Carlos Fernando Carlim Pinto

Orientador:

Prof. Julio Cesar Boscher Torres, D.Sc.

Co-orientador:

Prof. José Gabriel Rodriguez Carneiro Gomes, Ph.D.

Examinador:

Profa. Mariane Rembold Petraglia, Ph.D.

DEL

Junho de 2009

Dedicatória

A Deus.

Agradecimentos

À minha família, principalmente a meus pais que nunca duvidaram de minha capacidade e sempre me apoiaram em minhas escolhas.

Resumo

Neste trabalho investigou-se o comportamento de alguns padrões de alvos e avaliou-se qual padrão apresentou os maiores índices de acerto para uma rede neural. Os padrões a serem reconhecidos são alvos pré-definidos utilizados no processo de calibração dos sistemas de dimensionamento por estereofotogrametria e de reconstrução de cenas 2D e 3D.

O processo de calibração desses sistemas requer a ação humana de relacionar o pixel central de cada alvo à sua posição espacial real, relativa a um ponto de espaço. Este é um processo demorado, que requer precisão e paciência do apurador, que está sujeito a erros devido à repetição de apurações e de similaridade dos alvos. A associação equívocada de um alvo invalida toda a calibração. Como forma de evitar tais erros e automatizar o processo de calibração, foram treinadas redes neurais capazes de identificar, em uma imagem, os pixels centrais dos alvos, para que uma rotina possa, automaticamente, associá-los aos pontos do objeto de calibração.

A melhor rede projetada apresentou um desempenho com erro próximo de zero em seu treinamento, considerando um pré-processamento com Análise de Componentes Principais (PCA) e um pós-processamento de busca de centros de classes por “clusterização”.

Palavras-Chave

- Processamento de Imagens
- Rede Neural
- Análise de Componentes Principais (PCA)
- “Clusterização”
- Estereofotogrametria

Sumário

DEDICATÓRIA	2
AGRADECIMENTOS	3
RESUMO	4
PALAVRAS-CHAVE	5
SUMÁRIO	6
ÍNDICE DE FIGURAS.....	8
ÍNDICE DE TABELAS.....	12
GLOSSÁRIO E ABREVIATURAS.....	13
CAPÍTULO 1	14
INTRODUÇÃO.....	14
1.1 Motivação.....	14
1.2 Objetivos	15
1.3 Metodologia	16
CAPÍTULO 2	18
FUNDAMENTOS TEÓRICOS	18
2.1 Estereofotogrametria	18
2.2 Redes Neurais	18
2.2.1 Topologia	20
2.2.2 Taxa de Aprendizagem.....	21
2.2.3 Treinamento	22
2.2.4 <i>Backpropagation</i>	22
2.3 “Clusterização”	24
2.4 Análise de Componentes Principais.....	25

CAPÍTULO 3	26
PROJETO DAS REDES	26
3.1 Topologia.....	26
3.2 Tipos de Redes	26
3.3 Seleção de dados de entrada.....	32
3.4 Treinamento das Redes.....	33
3.5 Teste das redes.....	39
CAPÍTULO 4	44
TESTES E RESULTADOS	44
4.1 Redes Treinadas e Resultados	44
4.1.1 Teste 1	45
4.1.2 Teste 2	53
CAPÍTULO 5	60
CONCLUSÃO	60
REFERÊNCIAS BIBLIOGRÁFICAS	63
APÊNDICE A.....	64
APÊNDICE B.....	90

Índice de Figuras

FIGURA 1 - VISÃO GERAL DO PROJETO.....	16
FIGURA 2 - ESTRUTURA DE UM NEURÔNIO.....	19
FIGURA 3 - REDE DIRETA.....	20
FIGURA 4 - PROPAGAÇÃO DA ENTRADA.....	23
FIGURA 5 - AJUSTE DOS PESOS.....	24
FIGURA 6 – A) “X”; B)“CÍRCULO; C)“VASCO”.....	27
FIGURA 7 – FOTO DA GRADE CONTENDO O ALVO "X".....	28
FIGURA 8 - FOTO DA GRADE CONTENDO O ALVO "CÍRCULO".....	28
FIGURA 9 - FOTO DA GRADE CONTENDO O ALVO "VASCO".....	29
FIGURA 10 - GRADE VIRADA PARA A DIREITA.....	29
FIGURA 11 - GRADE VIRADA PARA A ESQUERDA.....	30
FIGURA 12 - CÂMERA NO MESMO NÍVEL DO GRADE.....	30
FIGURA 13 - CÂMERA ACIMA NO NÍVEL DO GRADE.....	31
FIGURA 14 - IMAGEM GIRADA 15 GRAUS.....	31
FIGURA 15 - IMAGEM GIRADA 30 GRAUS.....	31
FIGURA 16 - ALVOS RECORTADOS A 3 METROS DA CÂMERA. (A) CÍRCULO; (B)"X"; (C) VASCO.....	32
FIGURA 17 - ALVOS RECORTADOS A 2 METROS DA CÂMERA. (A) CÍRCULO; (B)"X"; (C) VASCO.....	32
FIGURA 18 - ALVOS RECORTADOS A 1 METROS DA CÂMERA. (A) CÍRCULO; (B)"X"; (C) VASCO.....	33
FIGURA 19 - EXEMPLO DE NÃO ALVOS RECORTADOS.....	33
FIGURA 20 - ALVO ROTACIONADO. (A) 15 GRAUS; (B) 30 GRAUS.....	33
FIGURA 21 - CURVA DE APRENDIZAGEM.....	34
FIGURA 22 – HISTOGRAMA.....	34
FIGURA 23 - CURVA DE APRENDIZAGEM DA REDECP.....	36
FIGURA 24 - HISTOGRAMA DO TREINAMENTO DA REDECP.....	36

FIGURA 25 - CURVA DE TREINO DA REDEVP	37
FIGURA 26 - HISTOGRAMA DO TREINAMENTO DA REDEVP.....	37
FIGURA 27 - CURVA DE TREINAMENTO DA REDEXP	38
FIGURA 28 - HISTOGRAMA DE TREINO DA REDEXP	38
FIGURA 29 - GRÁFICO DO ERRO DO TESTE DE UMA REDE.....	39
FIGURA 30 - HISTOGRAMA DO TESTE DE UMA REDE.....	40
FIGURA 31 - CURVA DE ERRO REFERENTE AO TESTE DA REDECP	41
FIGURA 32 - HISTOGRAMA DO TESTE DA REDECP	41
FIGURA 33 - CURVA DE ERRO REFERENTE À REDEVP	42
FIGURA 34 - HISTOGRAMA DO TESTE DA REDEVP.....	42
FIGURA 35 - CURVA DE ERRO DO TESTE REFERENTE À REDEXP.....	43
FIGURA 36 - HISTOGRAMA DO TESTE DA REDEXP.....	43
FIGURA 37 - EXEMPLO DE IMAGEM COM OS ALVOS MARCADOS	44
FIGURA 38 - ALVOS IDENTIFICADOS PELA REDECP COM GRADE A 1 METRO.....	46
FIGURA 39 - ALVOS IDENTIFICADOS PELA REDECP COM GRADE A 2 METROS.....	47
FIGURA 40 - ALVOS IDENTIFICADOS PELA REDECP COM GRADE A 3 METROS COM ZOOM	47
FIGURA 41 - ALVOS IDENTIFICADOS PELA REDEVP COM GRADE A 1 METRO.....	48
FIGURA 42 - ALVOS IDENTIFICADOS PELA REDEVP COM GRADE A 2 METROS	49
FIGURA 43 - ALVOS IDENTIFICADOS PELA REDEVP COM GRADE A 3 METROS	49
FIGURA 44 - ALVOS IDENTIFICADOS PELA REDEXP COM GRADE A 1 METRO.....	50
FIGURA 45 - ALVOS IDENTIFICADOS PELA REDEXP COM GRADE A 2 METROS	51
FIGURA 46 - ALVOS IDENTIFICADOS PELA REDEXP COM GRADE A 3 METROS	51
FIGURA 47 – GRADE POSICIONADO A 2 METROS DA CÂMERA.....	53
FIGURA 48 - GRADE POSICIONADA A 1 METRO DA CÂMERA.....	56
FIGURA 49 - ZOOM DA GRADE A 1 METRO DA CÂMERA	57
FIGURA 50 - GRADE A 2 METROS DA CÂMERA	57
FIGURA 51 - GRADE A 3 METROS DA CÂMERA	58
FIGURA 52 - GRADE A 1 METRO DA CÂMERA VIRADA PARA A DIREITA	58

FIGURA 53 - GRADE A 1 METRO DA CÂMERA VIRADA PARA A ESQUERDA.....	59
FIGURA 1 - CURVA DE APRENDIZAGEM DA REDECP1M.....	64
FIGURA 2 - HISTOGRAMA DO TREINO DA REDECP1M.....	64
FIGURA 3 - CURVA DE APRENDIZAGEM DA REDECP2M.....	65
FIGURA 4 - HISTOGRAMA DO TREINAMENTO DA REDECP2M.	65
FIGURA 5 - CURVA DE APRENDIZAGEM DA REDECP3M.....	66
FIGURA 6 - HISTOGRAMA DO TREINAMENTO DA REDECP3M	66
FIGURA 7 - CURVA DE APRENDIZAGEM DA REDECP	67
FIGURA 8 - HISTOGRAMA DO TREINAMENTO DA REDECP	67
FIGURA 9 - CURVA DE APRENDIZAGEM DA REDEVP1M.....	68
FIGURA 10 - HISTOGRAMA DO TREINAMENTO DA REDEVP1M	68
FIGURA 11 - CURVA DE TREINAMENTO DA REDEVP2M	69
FIGURA 12 - HISTOGRAMA DO TREINAMENTO DA REDEVP2M	69
FIGURA 13 - CURVA DE TREINAMENTO DA REDEVP3M	70
FIGURA 14 - HISTOGRAMA DO TREINAMENTO DA REDEVP3M	70
FIGURA 15 - CURVA DE TREINO DA REDEVP	71
FIGURA 16 - HISTOGRAMA DO TREINAMENTO DA REDEVP.....	71
FIGURA 17 - CURVA DE TREINAMENTO DA REDEVP2E3M.....	72
FIGURA 18 - HISTOGRAMA DE TREINAMENTO DA REDEVP2E3M	72
FIGURA 19 - CURVA DE TREINAMENTO DA REDEXP1M	73
FIGURA 20 - HISTOGRAMA DO TREINO DA REDEXP1M	73
FIGURA 21 - CURVA DE TREINO DA REDEXP2M	74
FIGURA 22 - HISTOGRAMA DO TREINO DA REDEXP2M	74
FIGURA 23 - CURVA DE TREINAMENTO DA REDEXP3M	75
FIGURA 24 - HISTOGRAMA DE TREINO DA REDEXP3M.....	75
FIGURA 25 - CURVA DE TREINAMENTO DA REDEXP	76
FIGURA 26 - HISTOGRAMA DE TREINO DA REDEXP	76
FIGURA 27 - GRÁFICO DE ERRO REFERENTE AO TESTE DA REDECP1M	77

FIGURA 28 - HISTOGRAMA DO TESTE DA REDECP1M	77
FIGURA 29 - CURVA DE ERRO DO TESTE REFERENTE À REDECP2M	78
FIGURA 30 - HISTOGRAMA DO TESTE DA REDECP2M	78
FIGURA 31 - CURVA DE ERRO REFERENTE AO TESTE DA REDECP3M	79
FIGURA 32 - HISTOGRAMA DO TESTE DA REDECP3M	79
FIGURA 33 - CURVA DE ERRO REFERENTE AO TESTE DA REDECP	80
FIGURA 34 - HISTOGRAMA DO TESTE DA REDECP	80
FIGURA 35 - CURVA DE ERRO REFERENTE À REDEV1M	81
FIGURA 36 - HISTOGRAMA DO TESTE DA REDEV1M	81
FIGURA 37 - CURVA DE ERRO REFERENTE À REDEV2M	82
FIGURA 38 - HISTOGRAMA DO TESTE DA REDEV2M	82
FIGURA 39 - CURVA DE ERRO REFERENTE À REDEV3M	83
FIGURA 40 - HISTOGRAMA DO TESTE DA REDEV3M	83
FIGURA 41 - CURVA DE ERRO REFERENTE À REDEV	84
FIGURA 42 - HISTOGRAMA DO TESTE DA REDEV	84
FIGURA 43 - CURVA DE ERRO REFERENTE À REDEV2E3M	85
FIGURA 44 - HISTOGRAMA DO TESTE DA REDEV2E3M	85
FIGURA 45 - CURVA DE ERRO DO TESTE REFERENTE À REDEXP1M	86
FIGURA 46 - HISTOGRAMA DO TESTE DA REDEXP1M	86
FIGURA 47 - CURVA DE ERRO DO TESTE REFERENTE À REDEXP2M	87
FIGURA 48 - HISTOGRAMA DO TESTE DA REDEXP2M	87
FIGURA 49 - CURVA DE ERRO DO TESTE REFERENTE À REDEXP3M	88
FIGURA 50 - HISTOGRAMA DO TESTE DA REDEXP3M	88
FIGURA 51 - CURVA DE ERRO DO TESTE REFERENTE À REDEXP	89
FIGURA 52 - HISTOGRAMA DO TESTE DA REDEXP	89

Índice de Tabelas

TABELA 1- TIPOS DE ALVO X Ω	52
--	----

Glossário e Abreviaturas

- **Neurônios**

Unidades de processamento da informação fundamental para a operação da Rede Neural Artificial. As operações realizadas pelos neurônios baseiam-se no processamento apenas de dados locais, isto é, apenas nas entradas recebidas por suas conexões. O neurônio é composto basicamente de duas unidades, o somador e a função de ativação.

- **Pixel**

Pixel é a menor unidade em uma imagem digital. Um conjunto de milhares de pixels forma uma imagem inteira. O termo megapixel é muito utilizado em câmeras digitais.

- **JPEG**

JPEG (*Joint Photographic Experts Group*) é um método de compactação de imagens com perdas. Os algoritmos são baseados na transformada DCT (*Discrete Cosine Transformation*), para descartar as partes menos significativas da imagem em termos de como ela é percebida pelo olho humano. Como é uma compactação com perdas, em cada edição haverá perda de dados. É o formato mais utilizado, atualmente, em câmeras digitais devido ao seu alto poder de compressão e conseqüente redução de espaço para armazenamento.

- **BMP**

BMP (bitmap) é o formato de imagem de bitmap padrão do Windows em computadores compatíveis com esse sistema operacional e com o DOS. Não há compressão com perdas e se baseia no mapeamento de pixels.

Capítulo 1

Introdução

1.1 Motivação

As principais reservas brasileiras de petróleo encontram-se em águas profundas e ultra profundas, dificultando o acesso de mergulhadores para a inspeção das instalações de produção, localizadas no fundo do mar, sendo necessária a utilização de imagens obtidas a partir de câmeras para realizar esta verificação.

A partir das imagens 2D, pelo processo de estereofotogrametria, o cenário real, em 3D, é reconstituído. E os objetos presentes na cena são dimensionados com auxílio de parâmetros de calibração destas câmeras. A calibração é feita pelo mapeamento dos pixels da imagem da cena 3D em coordenadas 2D. Atualmente este processo é feito remotamente por um operador. No intuito de tornar o processo de mapeamento automático, o uso de técnicas de redes neurais se mostra bastante eficiente.

Dois sistemas desenvolvidos no PADS (Laboratório de Processamento Digital e Analógico de sinais da UFRJ) necessitam de um processo de calibração, seja para estereofotogrametria ou para reconstruir objeto em cenas 2D e 3D.

Neste processo, diversos alvos – pixels centrais de uma figura padrão – precisam ser associados a pontos do espaço real tridimensional (x,y,z) . Cada alvo corresponde a um ponto no espaço.

De posse de uma lista de pontos associados (mapeamento das coordenadas da imagem para coordenadas do espaço 3D), uma rotina não-linear de minimização de erro quadrático atualiza um conjunto de parâmetros utilizados para realizar a projeção do espaço 3D (real) no espaço de imagem (2D) [Silva, 2003].

No projeto de final de curso de Juliana C. Achar, “Reconhecimento de padrões de calibração em estereofotogrametria através de redes neurais”, foi concluído que as técnicas de redes neurais podem ser usadas na calibração de câmeras em estereofotogrametria.

Nesse trabalho, verificou-se que o formato do alvo utilizado produzia diversos erros de identificação. Neste sentido, sugeriu-se uma melhor avaliação do tipo de alvo para a redução dos erros de identificação. Além disso, devido aos erros, não foi possível a elaboração de uma rotina que associasse diretamente os pixels identificados às posições (x,y,z) do espaço real.

1.2 Objetivos

Este trabalho tem como objetivo investigar alguns padrões de alvos e avaliar qual padrão apresenta os maiores índices de acerto para uma rede neural. Essa rede, então deverá ser capaz de identificar os alvos em uma imagem desconhecida e, através de pós-processamento de dados, associar os pixels centrais dos alvos a pontos (x,y,z) do espaço real.

1.3 Metodologia

Para alcançar os objetivos descritos acima, as etapas descritas são apresentadas na figura 1 devem ser realizadas com extrema cautela a fim de obter o menor erro possível nos resultados finais. Além disso, algumas rotinas foram criadas para automatizar alguns procedimentos.

A figura 1 é apresentada um diagrama das etapas seguidas durante o projeto.

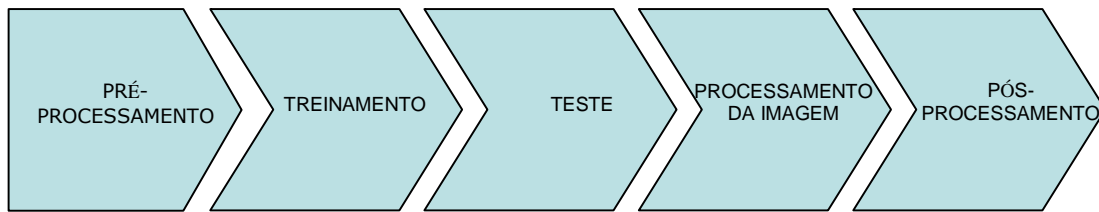


Figura 1 - Visão geral do projeto

Pré-processamento – Nesta etapa foram feitas as aquisições dos dados para serem usados no treinamento e no teste da rede criada. Estes dados foram separados cuidadosamente em dois conjuntos, um para treinamento e o outro para teste.

Treinamento – Consistiu no treinamento da rede para o reconhecimento dos padrões desejados.

Teste – os testes permitiram avaliação do desempenho das redes criadas e a escolha do “melhor” tipo de alvo. Este foi determinado pela observação e comparação das imagens após serem processadas por suas respectivas redes.

Para cada imagem foi determinada a razão

$$R = \frac{TOTAL_DE_ALVOS_CERTOS_IDENTIFICADOS}{TOTAL_DE_ALVOS_IDENTIFICADOS}$$

e definida como “melhor alvo” o que apresentar maior,

$$\Omega = \sum_{i=1}^3 R_i ,$$

tal que “ R_i ” é a distância em metros da câmera até a grade de alvos.

Deve ser definido como “TOTAL DE ALVOS CERTOS IDENTIFICADOS”, a soma dos alvos, padrões presentes na grade, com um pixel pintado de vermelho sobre o mesmo. E definido como “TOTAL DE ALVOS IDENTIFICADOS”, qualquer pixel pintado de vermelho na imagem.

Processamento da Imagem – Nesta etapa a rede processa uma imagem contendo os padrões (“alvos”), e fornece como saída as coordenadas dos pixels identificados.

Pós-processamento – Após o processamento da imagem pela rede, é necessário que os dados de saída sejam tratados, eliminando os falsos positivos e associando as coordenadas 2D da imagem processada a pontos 3D da cena real.

Capítulo 2

Fundamentos Teóricos

2.1 Estereofotogrametria

A fotogrametria permite realizar medições a partir de pares de fotografias, ou imagens, capturadas simultaneamente com um pequeno deslocamento de posição de uma câmera.

Assim, é possível estimar as coordenadas (3D) no espaço a partir de um par de imagens estéreo (2D).

O processo consiste em associar um conjunto de alvos – pixels centrais de uma figura padrão – a pontos do espaço 3D (cada alvo representa um ponto no espaço). A partir destes pontos, uma rotina não linear de minimização de erro quadrático atualiza um conjunto de parâmetros utilizados para realizar a projeção do espaço 3D no espaço da imagem 2D.[3]

2.2 Redes Neurais

As Redes Neurais, mais precisamente Redes Neurais Artificiais (RNA), são modelos matemáticos de processamento de dados inspirados no funcionamento dos neurônios presentes no cérebro. Tais redes, assim como um sistema nervoso biológico inteligente, adquirem conhecimento por aprendizado e são compostas por neurônios.

O neurônio é a unidade de processamento da informação fundamental para a operação da RNA. A Figura 2 apresenta a estrutura básica de um neurônio. Ele é composto de três elementos básicos:

- **Sinapses**, que são responsáveis por associar a cada entrada um peso, que determina o efeito da mesma sobre o neurônio. No caso da Figura 2, para cada uma das N entradas de dados X_n , com $n=1, \dots, N$, há um peso W_n associado.
- **Somador**, que combina linearmente os sinais de entrada e seus respectivos pesos.
- **Função de Ativação**, que determina o novo valor do estado de ativação, isto é, determina o nível de ativação do neurônio artificial.

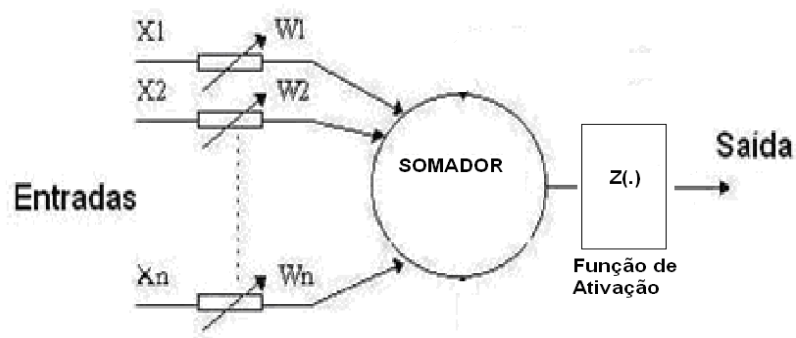


Figura 2 - Estrutura de um neurônio

2.2.1 Topologia

- Rede direta (“*feedforward*”) – São redes em que o sinal é propagado em um único sentido, da entrada diretamente para a saída, isto é, não possuem ciclos. Como pode ser observado na Figura 3.

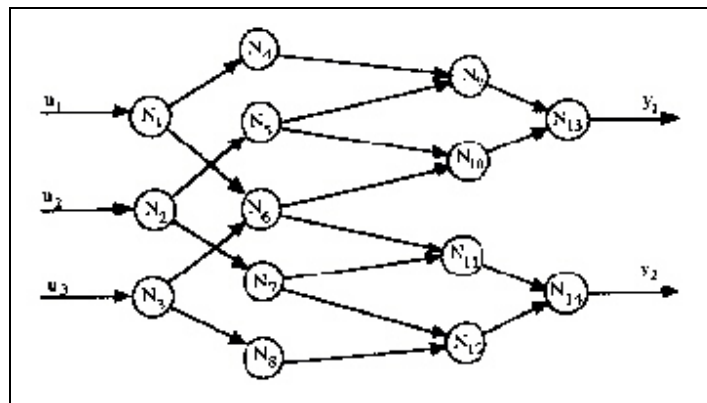


Figura 3 - Rede Direta

- Rede com ciclos (“*feedback*”) – São redes em que o sinal se propaga em ambos os sentidos, graças à introdução de realimentação na rede.
- Rede simétrica – São casos particulares de rede em ciclos, em que a configuração dos ciclos é arranjada simetricamente.

2.2.2 Taxa de Aprendizagem

Gradiente Descendente é um método para se determinar mínimo local através de ajustes sucessivos do peso do vetor W na direção oposta ao vetor gradiente da função custo. O algoritmo pode ser descrito formalmente por:

$$W(n+1) = W(n) - \eta \nabla \mathcal{E}(w)$$

$$\Delta W(n) = \eta g(n)$$

O parâmetro η , parâmetro de taxa de aprendizagem, define o tamanho do passo e conseqüentemente a velocidade de convergência, e $g(n)$ é a direção oposta do vetor gradiente no ponto $W(n)$.

Quanto menor *eta*, menores são as variações do vetor peso W e mais suave é a trajetória da convergência para o mínimo. Contudo, o tempo de aprendizagem é maior. Conforme se aumenta *eta*, mais rápida é a aprendizagem, já que as variações do vetor W são maiores. Contudo, a rede pode se tornar instável e não convergir.

Para evitar a instabilidade quando se utiliza um *eta* “grande”, é incluído à regra delta um termo de momento α , denominado “constante de momento”, tal que $0 \leq \alpha < 1$. Como pode ser observado a seguir, na equação definida como Regra Delta Generalizada:

$$\Delta W(n) = \alpha \Delta W(n-1) + \eta g(n)$$

A constante de momento diminui as variações de direção quando o peso é atualizado, conseqüentemente evitando que o processo de aprendizagem convirja para um mínimo local.

2.2.3 Treinamento

O aprendizado de uma RNA é definido como a determinação dos pesos para as entradas fornecidas durante o processo de treinamento, o qual é realizado de época em época. Uma Época corresponde a uma apresentação completa do conjunto de treinamento da RNA.

Definição: Época corresponde a uma apresentação completa do conjunto de treinamento inteiro à RNA.

Os pesos são definidos pelos algoritmos de treinamento, dentre os muitos existentes o utilizado neste projeto foi o *Error Backpropagation*. O treinamento pode ser realizado por um dos dois métodos apresentados a seguir:

- **Sequencial** – Ajusta o peso das entradas após a apresentação de cada dado de treinamento. Isto é, o peso é atualizado a cada iteração da entrada pelo erro na saída.
- **Batelada** – Ajusta o peso das entradas a cada época. Isto é, o peso é atualizado após a apresentação de todos os conjuntos de treinamento.

O modo sequencial é computacionalmente mais rápido que a batelada. Esta afirmação pode ser fortemente confirmada quando utiliza-se um conjunto de dados de treinamento com muitos elementos e altamente redundante.

2.2.4 *Backpropagation*

Backpropagation é um algoritmo de treinamento de redes multicamadas, o qual se baseia no Aprendizado Supervisionado por Correção de Erros. O algoritmo pode ser

considerado como uma Regra Delta Generalizada para redes diretas com mais de duas camadas. De maneira bem simplificada, a retroprogramação consiste em usar o erro determinado pela diferença entre a saída atual e a desejada e modificar o peso da entrada para que este erro seja zero.

2.2.4.1 Algoritmo

Abaixo é apresentado o algoritmo em duas etapas:

1. Propagação – Após a apresentação do padrão de entrada, a resposta de uma entrada é propagada como entrada da próxima camada, até a camada de saída, onde, a partir da resposta de saída, o erro é calculado, conforme se observa na Figura 4.

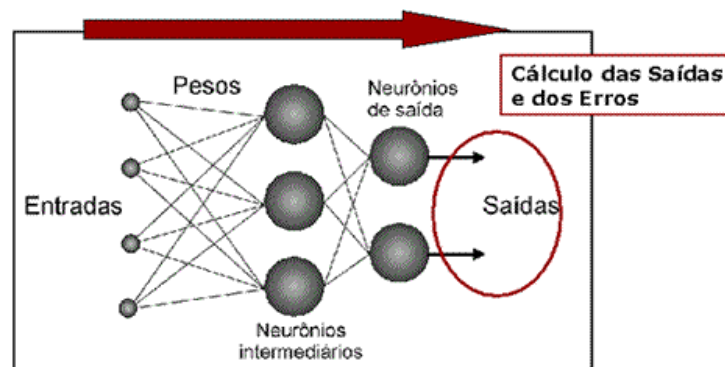


Figura 4 - Propagação da entrada

2. Retropropagação: A partir da camada de saída os pesos das entradas das camadas são alterados até a camada de entrada, como mostrado na Figura 4.

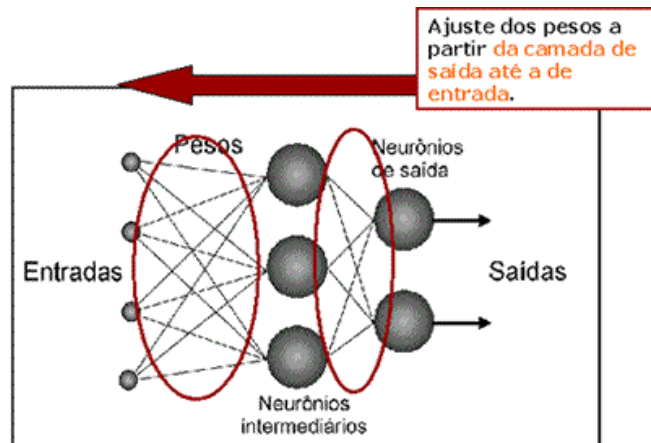


Figura 5 - Ajuste dos pesos

2.2.4.2 Execução da Rede

O algoritmo *Backpropagation* foi utilizado neste projeto para o treino de rede neural. Os passos de treinamento são os descritos no item 2.2.4.1.

A execução da rede consiste nas seguintes etapas:

1. Sinais são apresentados à entrada;
2. Cada sinal é multiplicado por um peso, que indica a sua influência na saída do neurônio;
3. É feita a soma ponderada dos sinais, que produz um nível de atividade;
4. Se este nível de atividade exceder o limite pré-determinado, o neurônio identifica a entrada como “alvo”. Do contrário, decide por “não alvo”.

2.3 “Clusterização”

O termo “Clusterização” é utilizado para definir a classificação não-supervisionada de dados, formando agrupamentos ou *clusters*. Ela representa uma das principais etapas de processos de análise de dados, denominada análise de clusters. [6]

A análise de clusters envolve, portanto, a organização de um conjunto de padrões (usualmente representados na forma de vetores de atributos ou pontos em um espaço multidimensional – espaço de atributos) em clusters, de acordo com alguma medida de similaridade. Intuitivamente, padrões pertencentes a um dado cluster devem ser mais “similares” entre si do que em relação a padrões pertencentes a outros clusters.

2.4 Análise de Componentes Principais

Também conhecida como Transformada de Karhunen-Loève, a Análise de Componentes Principais (PCA) maximiza a redução da variância das variáveis. Em síntese a PCA é um método que tem por finalidade básica a redução de dados a partir de combinações lineares das variáveis originais.

Esta é uma técnica de transformação de variáveis onde, primeiramente, são calculados os autovalores e autovetores de uma matriz de correlações de variáveis. Após ordenação o primeiro autovalor determinado corresponde à maior porcentagem da variabilidade total e assim sucessivamente. Com isto, é possível reduzir drasticamente as informações linearmente dependentes e redundantes na entrada da rede neural, agilizando sua performance.

Capítulo 3

Projeto das Redes

Este capítulo apresenta simplificadaamente a topologia da RNA utilizada, em seguida são apresentados os alvos de treinamento e por último são apresentados os resultados obtidos pelo processamento de cada rede treinada.

3.1 Topologia

O trabalho utilizou uma rede do tipo *perceptron* multicamadas com quatorze entradas e uma saída. Treinamento seqüencial com duas camadas, taxa de aprendizado de 0,01, momento constante igual a zero e um erro quadrático instantâneo dado pela expressão

$$E = \frac{1}{2} e^2, \text{ foi realizado.}$$

3.2 Tipos de Redes

O trabalho consistiu na definição de três tipos de alvos e para cada tipo foram utilizadas três distâncias diferentes de treinamento. Cada alvo gerou uma RNA, capaz de reconhecê-lo nas três distâncias.

A Figura 6 apresenta os três tipos de alvos, o primeiro é chamado de “X”, o segundo de “Círculo” e o terceiro de “Vasco”.

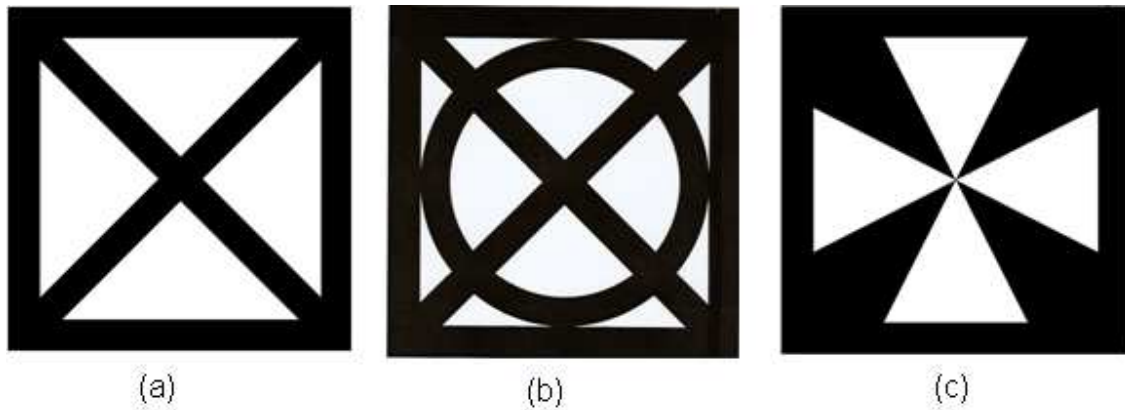


Figura 6 – a) “X”; b) “círculo; c) “vasco”

Foi criada uma grade de 50 cm x 50 cm onde fixaram-se os alvos. Para cada alvo a grade foi posicionada a 1, 2 e 3 metros da câmera, a qual estava configurada com uma resolução de 3.2 megapixels.

Utilizou-se uma janela de 31 pixels x 31 pixels, para selecionar o alvo nas imagens. O tamanho ideal da janela foi determinado empiricamente, pois estas dimensões permitiram selecionar satisfatoriamente os alvos nas três distâncias. Nas Figuras 7, 8 e 9 podem ser observadas as grades de cada alvo.

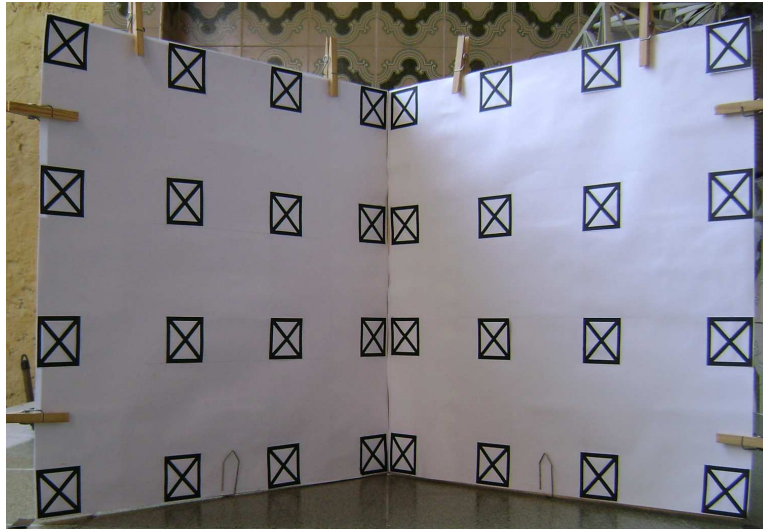


Figura 7 – Foto da grade contendo o alvo "X"

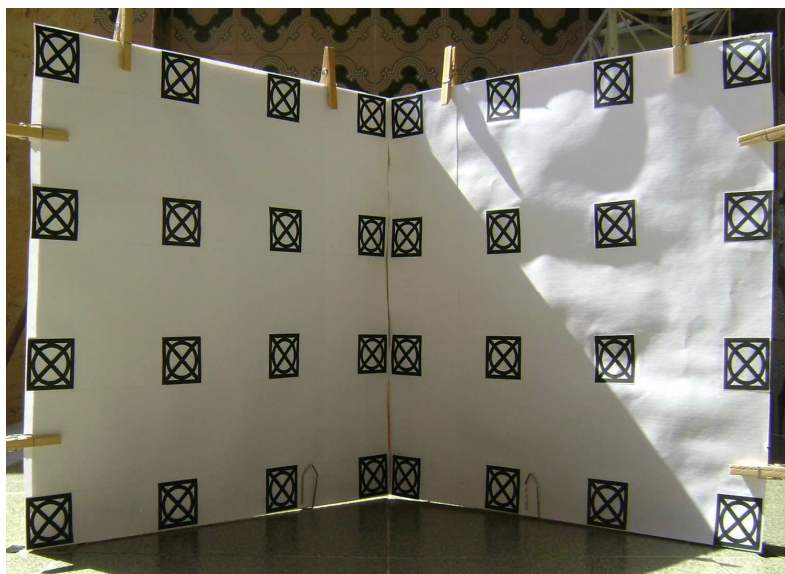


Figura 8 - Foto da grade contendo o alvo "círculo"

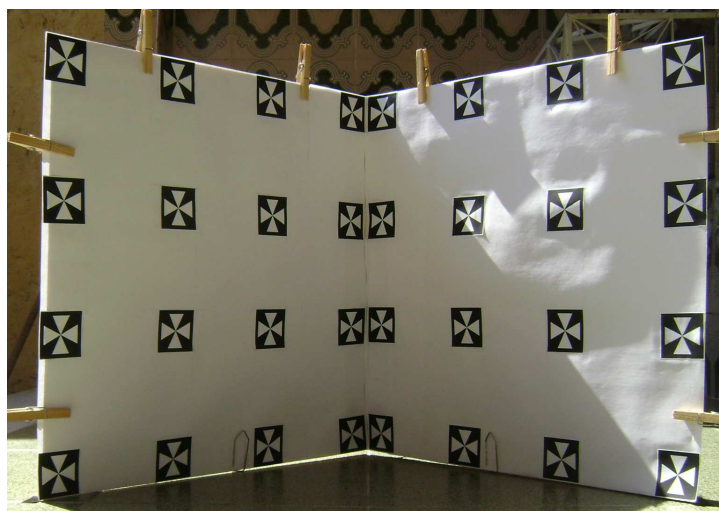


Figura 9 - Foto da grade contendo o alvo "Vasco"

Em cada distância a grade foi colocada em três posições diferentes, de frente (Figuras 7, 8 e 9), virada para a direita (figura 10) e virada para a esquerda (figura 11).

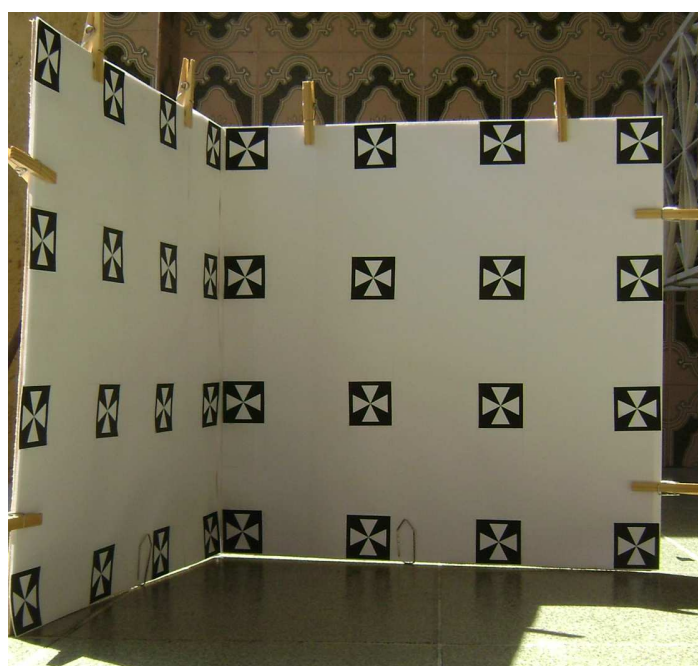


Figura 10 - Grade virada para a direita.

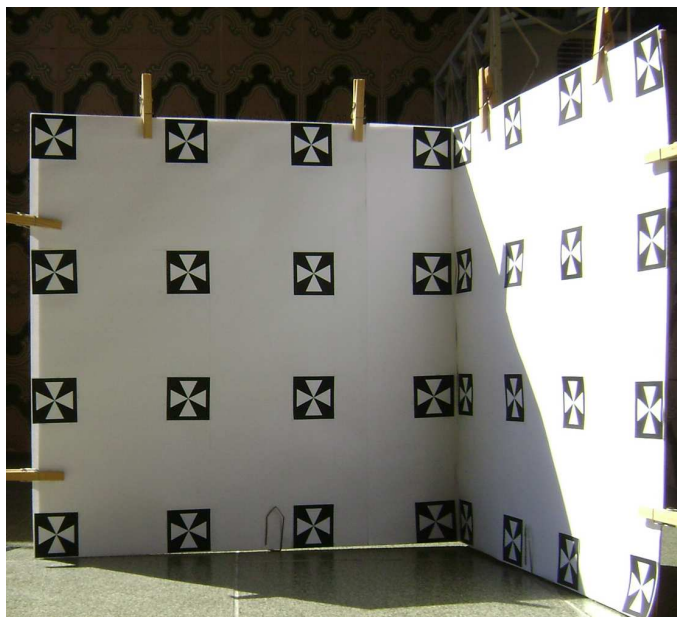


Figura 11 - Grade virada para a esquerda.

Para cada posição da grade, a câmera foi colocada em duas alturas diferentes: no mesmo nível da grade (figura 12) e acima da grade (figura 13). Cada foto foi girada para a direita em 15° (figura 14) e 30° (figura 15).

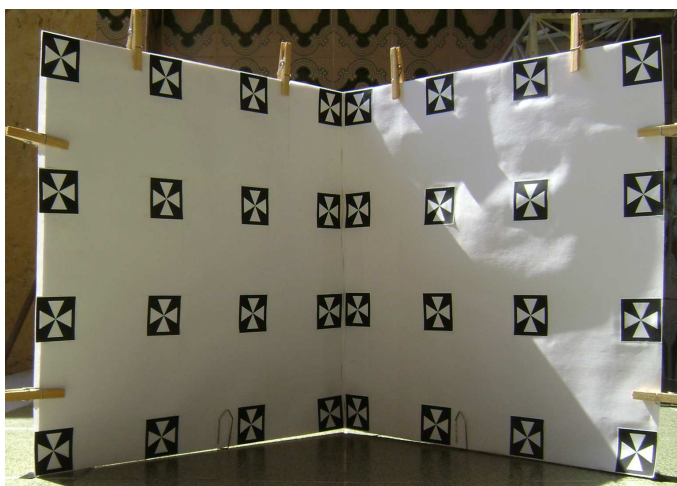


Figura 12 - Câmera no mesmo nível do grade.

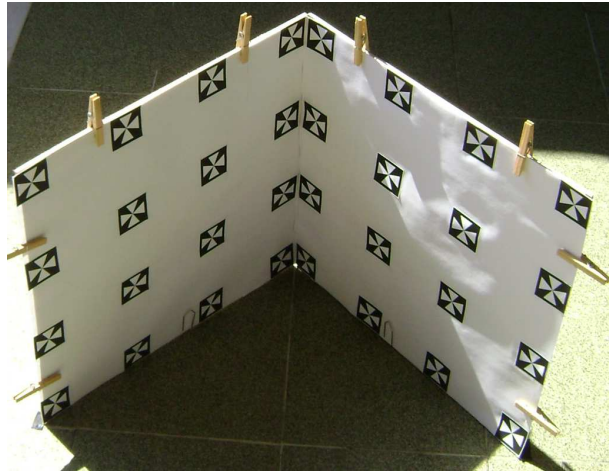


Figura 13 - Câmera acima no nível do grade.

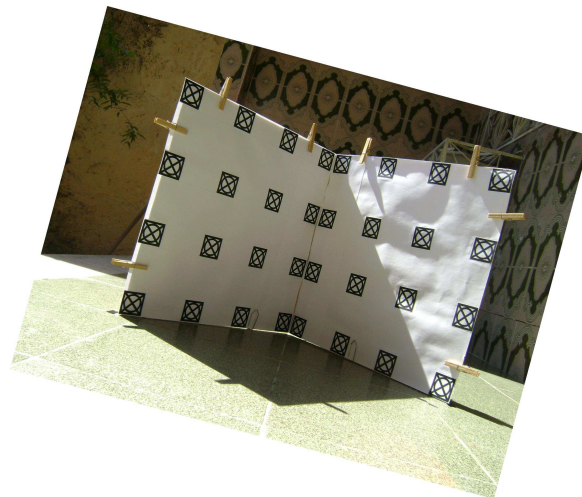


Figura 14 - Imagem girada 15 graus



Figura 15 - Imagem girada 30 graus

3.3 Seleção de dados de entrada

As fotos foram agrupadas por tipo de alvo e distância da câmera. De cada foto foram recortados dois tipos de padrões: “alvos” (Figuras 16, 17 e 18) e “não alvos” (Figura 19). Cada imagem gerada recebeu um nome específico, no qual as duas primeiras letras indicam se a imagem é alvo, “al”, ou não alvo, “na”, as duas letras seguintes indicam o tipo de alvo, “cp” para o círculo, “xp” para o X e “vp” para o Vasco. Em seguida aparece um número e uma letra indicando a distância da grade até a câmera, “1m” para 1 metro, “2m” para 2 metros e “3m” para 3 metros, os dois números seguintes indicam o número da imagem de origem. Os dois últimos números são sequenciais para cada imagem origem. Temos aqui um exemplo, alcp3m6201.bmp. Esta imagem é um alvo, do tipo círculo, estava a 3 metros de distancia da câmera e foi a primeira imagem recortada da foto 62. Os padrões foram recortados das imagens originais e também de versões rotacionadas de 15° e 30° no sentido horário, como mostrado na figura 20.

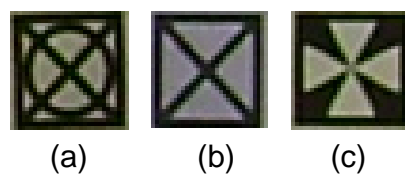


Figura 16 - Alvos recortados a 3 metros da câmera. (a) círculo; (b) "X"; (c) vasco

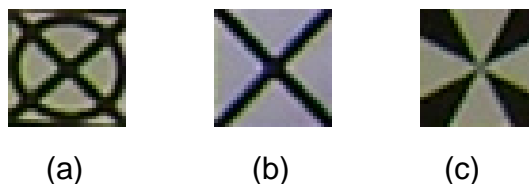


Figura 17 - Alvos recortados a 2 metros da câmera. (a) círculo; (b) "X"; (c) Vasco

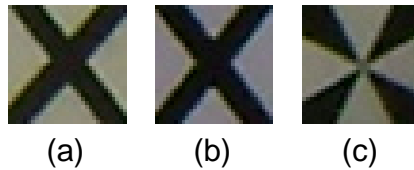


Figura 18 - Alvos recortados a 1 metros da câmera. (a) círculo; (b) "X"; (c) Vasco

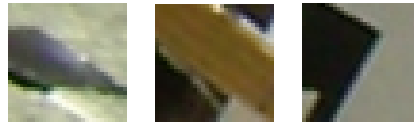


Figura 19 - Exemplo de não alvos recortados

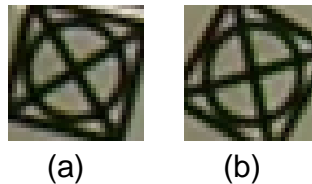


Figura 20 - Alvo Rotacionado. (a) 15 graus; (b) 30 graus

Cada tipo de “alvos” e “não alvos” foi agrupado por distância. Cada grupamento foi separado em dados para treino e dados para teste, tendo o cuidado de colocar no conjunto de teste imagens semelhantes às colocadas no conjunto de treinamento. Para que durante a etapa de teste a rede seja capaz de reconhecer as imagens apresentadas.

3.4 Treinamento das Redes

Os alvos e não alvos de treinamento foram separados por distância da grade, o que gerou uma rede para cada tipo de alvo e distância.

Durante o treinamento foi necessário a retirada de alguns dados de entrada, tanto no conjunto de alvos, como no conjunto de não alvos, pois estes estavam dificultando o aprendizado da rede. Tal fato pode ser observado pela análise da curva de aprendizagem

da rede, que é um gráfico do erro absoluto (eixo y) pelo número de iterações (eixo x) que avalia a evolução do erro. A figura 21 exemplifica uma dessas curvas, obtida após seleção cuidadosa dos alvos. Observa-se que ao longo do treinamento o erro tende a zero, significando um bom treinamento.

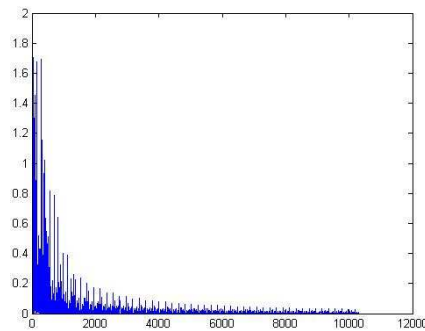


Figura 21 - Curva de aprendizagem

Foram plotados, também, histogramas (Figura 22), que permitem avaliar o limiar de separação dos padrões de saída da rede, isto é, a partir de qual valor a rede define o que é alvo e o que não é alvo. Observando-se o histograma da Figura 22, percebe-se um limiar em aproximadamente 0,8, isto é, o que foi identificado como “alvo” está acima de 0,8 e o que foi identificado como “não alvo” está abaixo de -0,8.

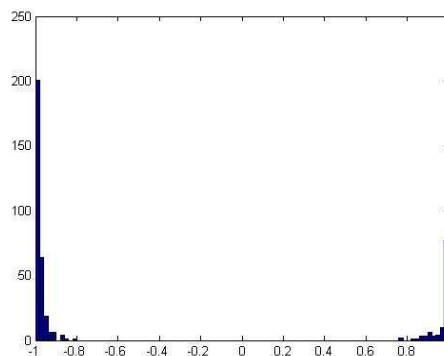


Figura 22 – Histograma

A seguir serão apresentadas as curvas de aprendizagem e os histogramas obtidos com o treinamento das redes.

Para se treinar uma rede capaz de identificar alvos em todas as distâncias foi necessária uma seleção prévia dos dados de entrada. Esta seleção foi feita individualmente para cada distância e a qualidade dos dados de entrada foi avaliada pelos gráficos que serão apresentados no Apêndice A.

No caso do alvo círculo, inicialmente, foi treinada uma rede contendo apenas as imagens recortadas da grade a 1 metro da câmera e a partir da observação da curva de aprendizagem e do histograma pôde-se identificar quais dados estavam dificultando o treinamento da rede. Tais dados foram retirados do conjunto de treinamento e a rede foi treinada novamente. Este processo foi repetido até obterem-se resultados que indicassem o bom treinamento da rede. O mesmo foi feito para as imagens recortadas da grade a 2 e 3 metros da câmera. Estes gráficos podem ser confirmados no Apêndice A. Após este procedimento os três conjuntos de “alvos” foram unidos em um único e uma nova rede foi treinada (denominada “redeCP”), cujo objetivo é identificar o alvo do tipo círculo nas três distâncias.

O mesmo procedimento descrito acima foi empregado para os alvos do tipo “Vasco”, gerando a “redeVP” e “X”, gerando a “redeXP”.

Alvo do tipo Círculo

Os gráficos da Figura 23 foram obtidos a partir do treinamento de uma rede tendo como entrada os alvos e não alvos do tipo círculo nas três distâncias da câmera. Esta rede foi nomeada de “redeCP”.

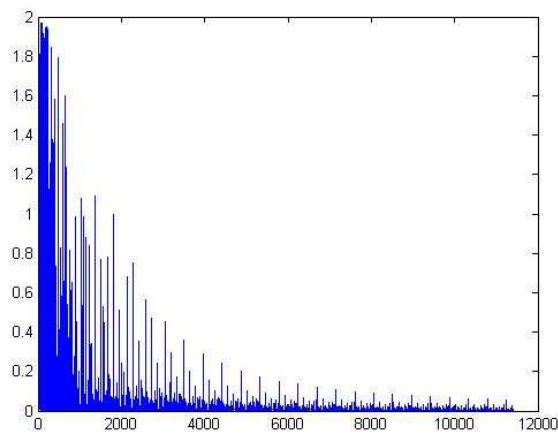


Figura 23 - Curva de aprendizagem da redeCP

O gráfico da Figura 24 demonstra que a rede conseguiu ser bem treinada para as três distâncias da grade.

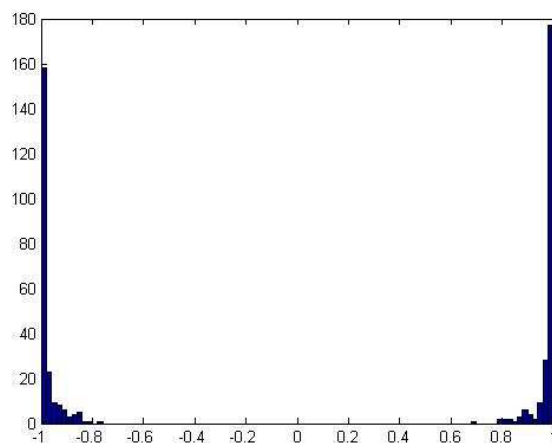


Figura 24 - Histograma do treinamento da redeCP

Alvo do tipo Vasco

Os gráficos abaixo foram obtidos a partir do treinamento de uma rede tendo como entrada os alvos e não alvos do tipo Vasco nas três distâncias da câmera. Esta rede foi nomeada de “redeVP”.

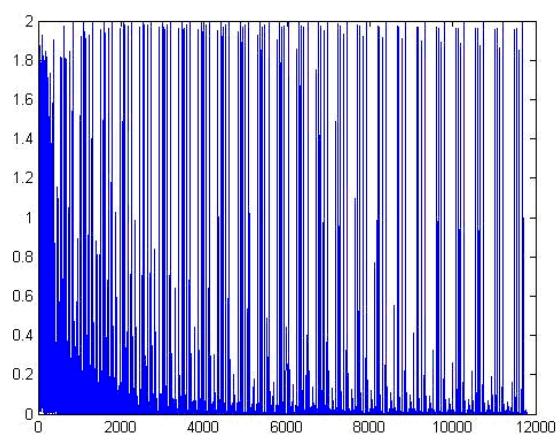


Figura 25 - Curva de treino da redeVP

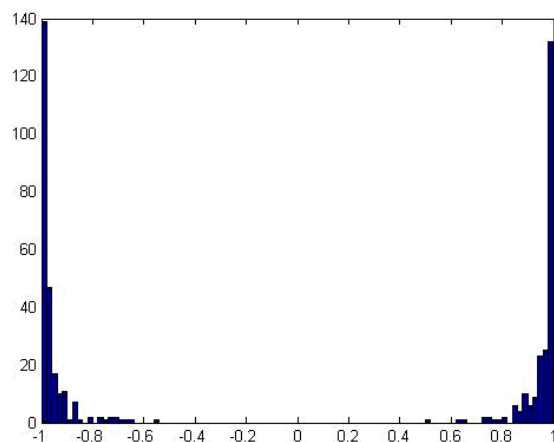


Figura 26 - Histograma do treinamento da redeVP

Da figura 25 observa-se que um conjunto de padrões não conseguiu ser treinado, pois o erro em cada época não diminuiu com o número de épocas. Este conjunto é formado pelos alvos localizados a 1 metro da câmera, conforme descrito no APENDICE A.

Alvo do tipo X

Os gráficos abaixo (Figuras 27 e 28) foram obtidos a partir do treinamento de uma rede tendo como entrada os alvos e não alvos do tipo X nas três distâncias da câmera. Esta rede foi nomeada de “redeXP”.

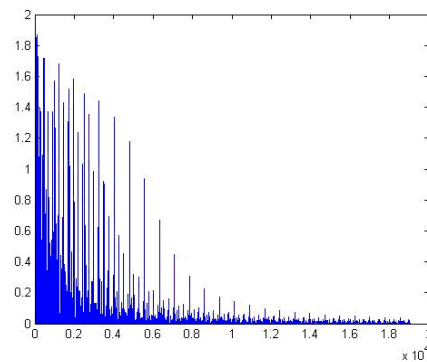


Figura 27 - Curva de treinamento da redeXP

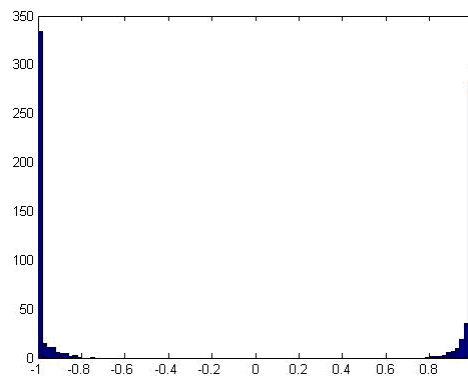


Figura 28 - Histograma de treino da redeXP

Os gráficos acima (Figuras 27 e 28) demonstram que a redeXP foi treinada para identificar o alvo nas três distâncias diferentes, chegando a um erro muito próximo de zero e um limiar de 0,8.

3.5 Teste das redes

O teste das redes foi realizado pela apresentação de padrões – conjunto de alvos e não alvos – nunca antes vistos pela rede e seu comportamento avaliado pelo gráfico de erro (Figura 29), que indica quando um alvo é identificado como não alvo, ou o contrario. Uma rede ótima seria a que obtivesse erro zero. Para todos os gráficos, a primeira metade de pontos refere-se aos alvos, enquanto a outra metade refere-se aos não alvos.

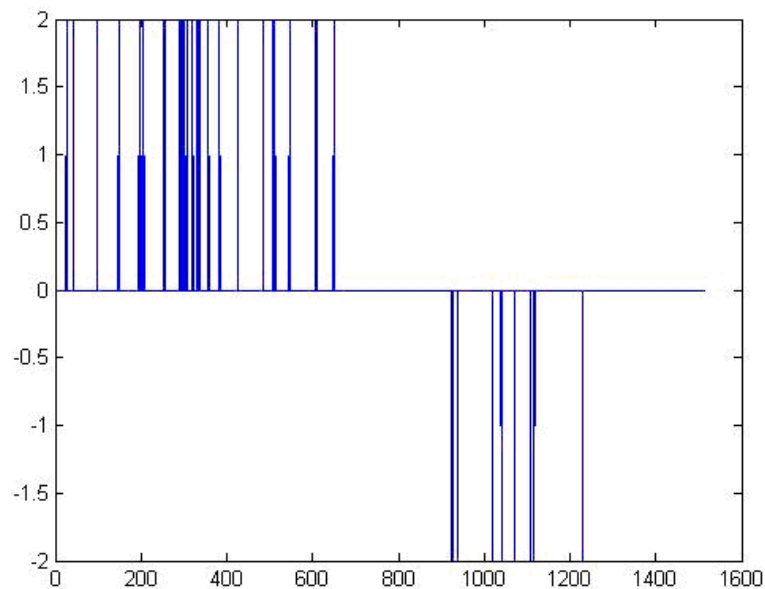


Figura 29 - Gráfico do erro do teste de uma rede

De acordo com a Figura 29, um pico positivo indica um alvo identificado como não alvo e um pico negativo indica um não alvo identificado como alvo.

Um histograma também é gerado ao fim rotina de teste (Figura 30), permitindo avaliar o grau de separação alcançado pela rede quando apresentada a dados nunca vistos.

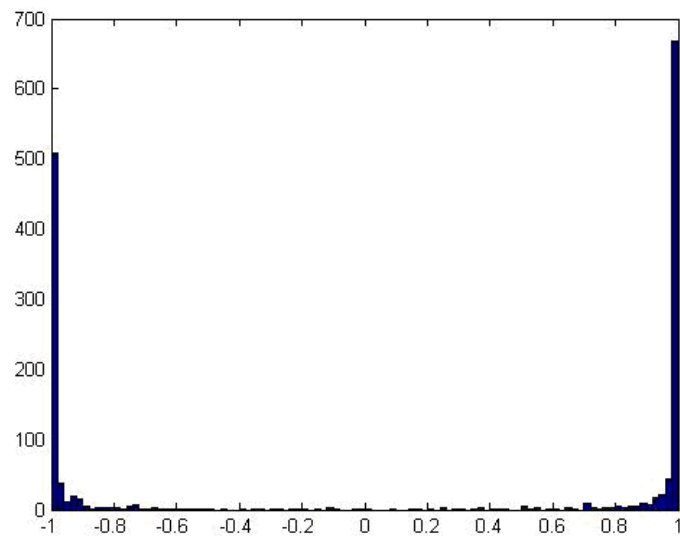


Figura 30 - Histograma do teste de uma rede

Alvo do tipo Círculo

Neste tópico será apresentado o teste realizado para a redeCP, tendo como entrada os alvos localizados nas três distâncias da câmera.

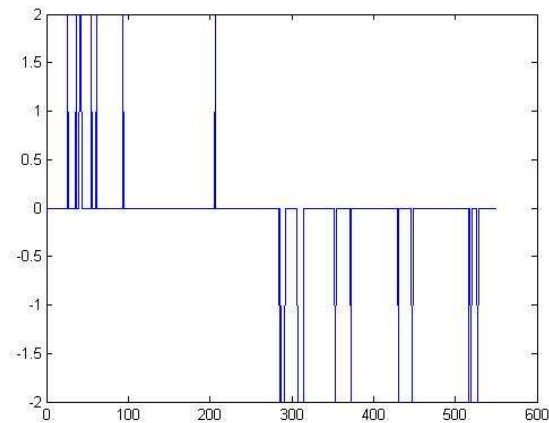


Figura 31 - Curva de erro referente ao teste da redeCP

A curva de erro da Figura 31 mostra que 8 alvos foram identificados como não alvos e 10 não alvos foram identificados como alvo.

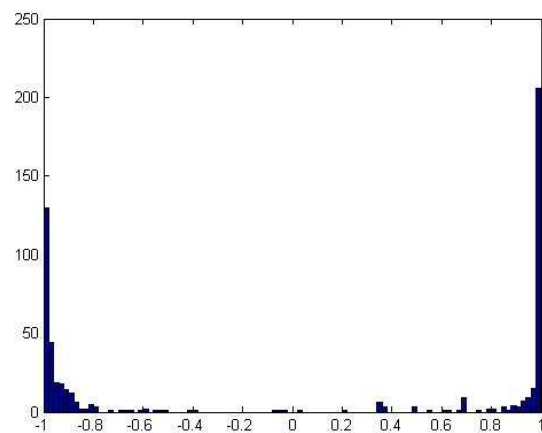


Figura 32 - Histograma do teste da redeCP

O histograma da Figura 32 apresentou muita dispersão em ambos os lados.

Alvo do tipo Vasco

Neste tópico será apresentado o teste realizado para a redeVP, tendo como entrada os alvos localizados nas três distâncias da câmera.

A curva de erro da Figura 33 mostra que dois alvos foram identificados como não alvo e um não alvo foi identificado como alvo. Apesar da rede tipo Vasco apresentar problemas durante o treinamento, seus teste apresentaram resultados positivos. Tal fato deve-se a presença no conjunto de treinamento de apenas de padrões para os quais a rede é capaz de identificar com precisão.

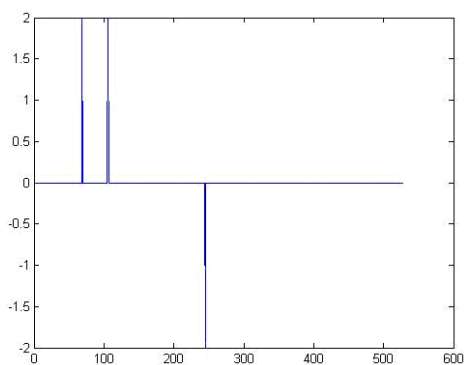


Figura 33 - Curva de erro referente à redeVP

Apesar dos pontos dispersos, o histograma da Figura 34 obteve uma separação satisfatória.

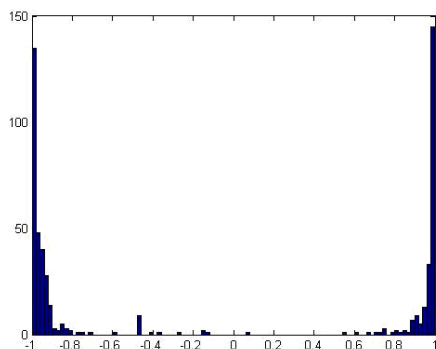


Figura 34 - Histograma do teste da redeVP

Alvo do tipo X

Neste tópico será apresentado o teste realizado para a redeXP, tendo como entrada os alvos localizados nas três distâncias da câmera.

A curva de erro da Figura 35 apresenta cinco alvos identificados como não alvos e um não alvo identificado como alvo.

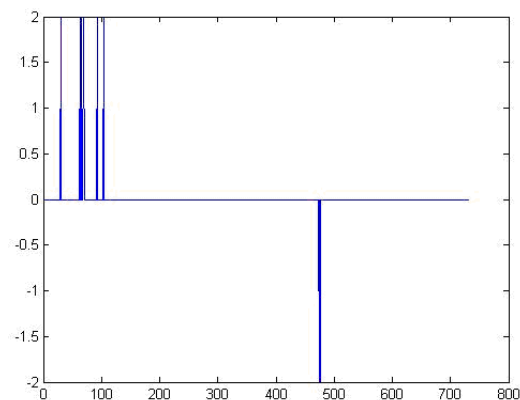


Figura 35 - Curva de erro do teste referente à redeXP

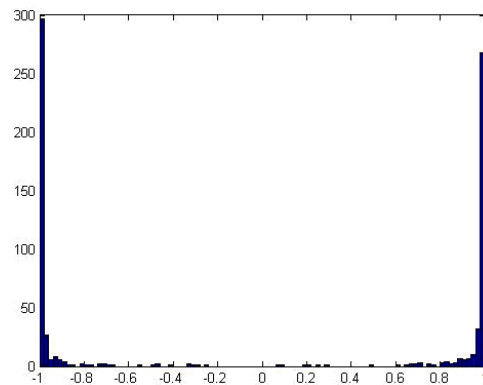


Figura 36 - Histograma do teste da redeXP

O histograma da Figura 36 mostra a maior concentração de pontos acima de 0,6, indicando que a rede separa os alvos dos não alvos, apesar de alguns erros.

Capítulo 4

Testes e Resultados

4.1 Redes Treinadas e Resultados

Neste capítulo serão apresentados os resultados dos testes realizados com as redes treinadas: RedeCP, RedeVP e RedeXP. Com a ajuda da rotina “processa_imagem”, as imagens da grade foram varridas pelas redes e pintadas de vermelho nos pixels identificados como alvos. Para os testes foram utilizadas apenas fotos tiradas com a grade de frente e no mesmo nível da câmera.

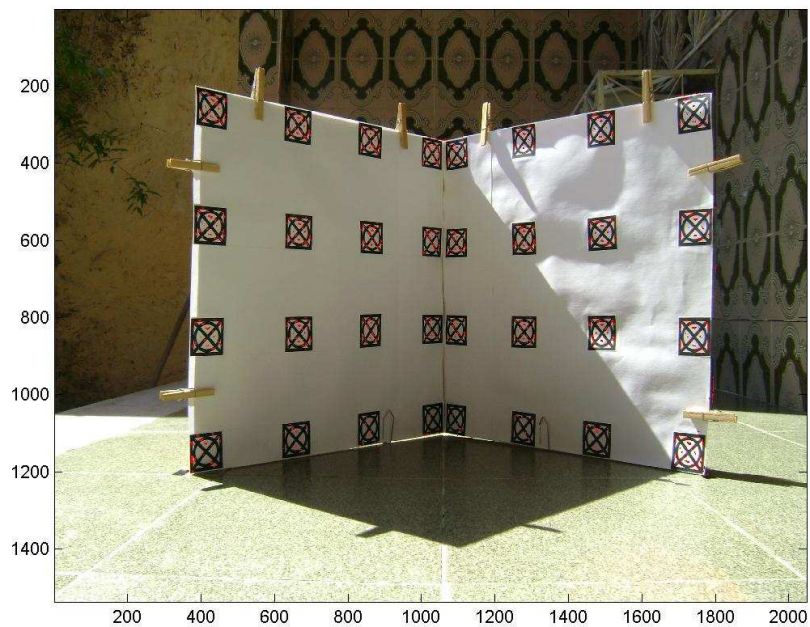


Figura 37 - Exemplo de imagem com os alvos marcados

4.1.1 Teste 1

Este teste tem como objetivo identificar qual o tipo de alvo mais indicado em um sistema de calibração, isto é, para qual alvo as redes cometeram menos erros durante o processamento das imagens. Para isso foram utilizadas as imagens das quais os dados de treinamento das redes foram retirados e calculado o Ω (definido no item 1.3) para cada rede.

Os resultados possibilitaram a identificação de alguns problemas a serem corrigidos, tais como, a “nuvem” de pontos em torno do pixel central identificado como alvo e os falsos alvos. Pode-se determinar também que o alvo do tipo Círculo, por ter apresentado o maior Ω , é o mais indicado para ser utilizado em um sistema de calibração, conforme será mostrado a seguir.

Por conveniência durante o teste 1 define-se o grupo (“nuvem”) de pontos como “alvo”. E todos os grupos foram circutados manualmente de verde para uma melhor visualização.

Alvo tipo Círculo

Neste item serão apresentados os resultados da varredura das imagens com o alvo do tipo Círculo pela redeCP. A rede obteve bons resultados, sendo capaz de identificar com eficiência os alvos com a grade posicionada nas três distancias com poucos erros.

Como pode ser observado na Figura 38 todos alvos foram pintados de vermelho, apresentando um $R_1 = 1$.

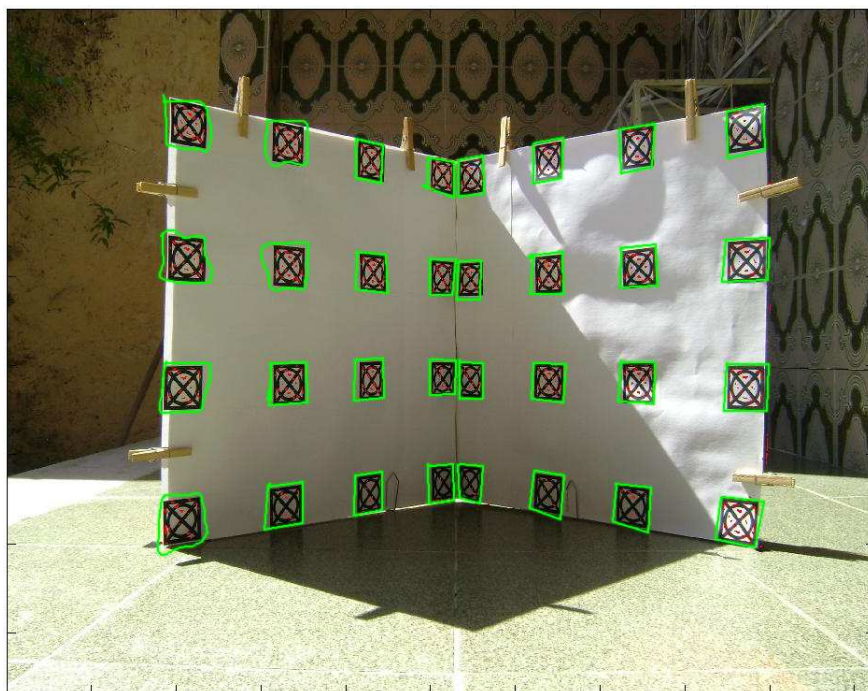


Figura 38 - Alvos identificados pela redeCP com grade a 1 metro

Na Figura 39 todos os alvos foram pintados de vermelho, contudo foram identificados três falsos alvos. Apesar de indicar três falsos alvos a rede continua mostrando-se eficiente na seleção dos alvos, com um $R_2 = 0,91$.

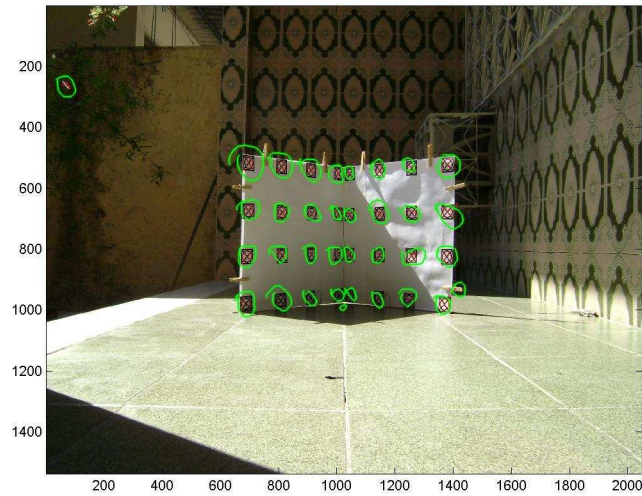


Figura 39 - Alvos identificados pela redeCP com grade a 2 metros

Com o objetivo de melhorar a visualização do resultado, a figura 40 mostra um zoom da imagem da grade a 3 metros. O resultado mostra que dois alvos não foram identificados e a rede selecionou quatro falsos alvos, com um $R_3 = 0,88$.

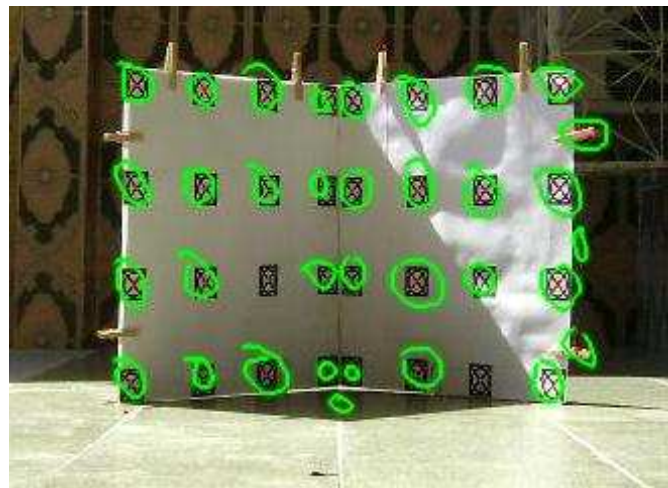


Figura 40 - Alvos identificados pela redeCP com grade a 3 metros com zoom

A partir da soma de R_1 , R_2 e R_3 calculou-se: $\Omega = R_1 + R_2 + R_3 = 2,79$.

Alvo tipo Vasco

Neste item serão apresentados os resultados da varredura das imagens com o alvo do tipo Vasco pela redeVP. Com a grade a 1 metro da câmera a rede obteve um bom resultado, com poucos erros e todos os alvos identificados. Porém com a grade posicionada a 2 e 3 metros os resultados mostraram que a rede não conseguiu diferenciar de maneira confiável alvos de não alvos.

A figura 41 mostra todos os alvos com a grade a 1 metro de distância da câmera e onze falsos alvos foram selecionados, gerando um $R_1 = 0,73$.

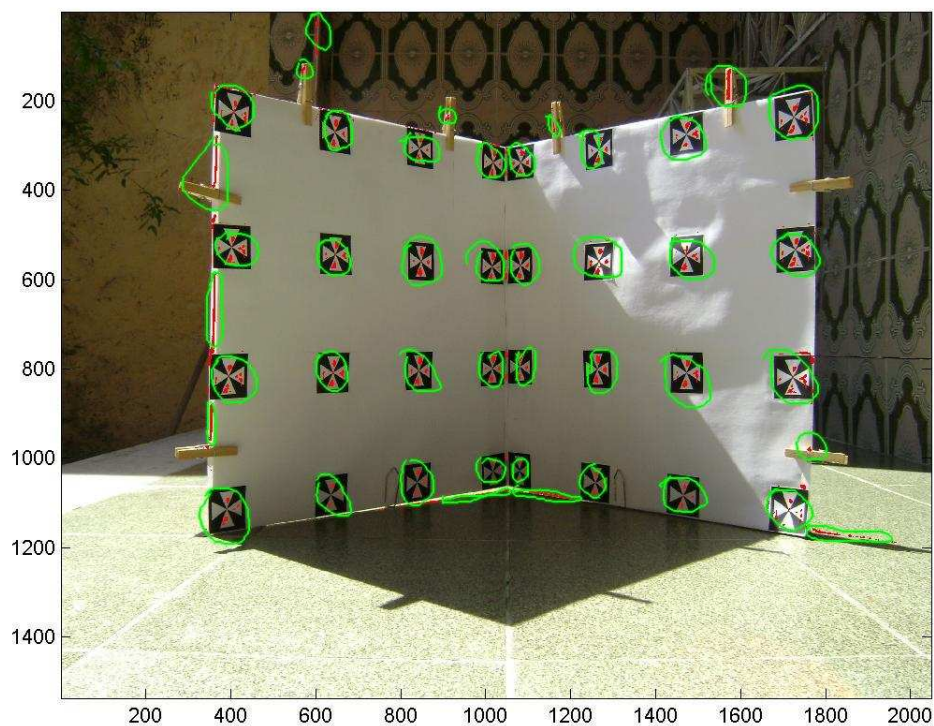


Figura 41 - Alvos identificados pela redeVP com grade a 1 metro

A figura 42 e 43 mostra que a rede identificou diversos falsos alvos e não marcou dois alvos com as grades posicionadas a 2 e 3 metros da câmera, respectivamente, apresentando $R_2 = 0,49$ e $R_3 = 0,56$.

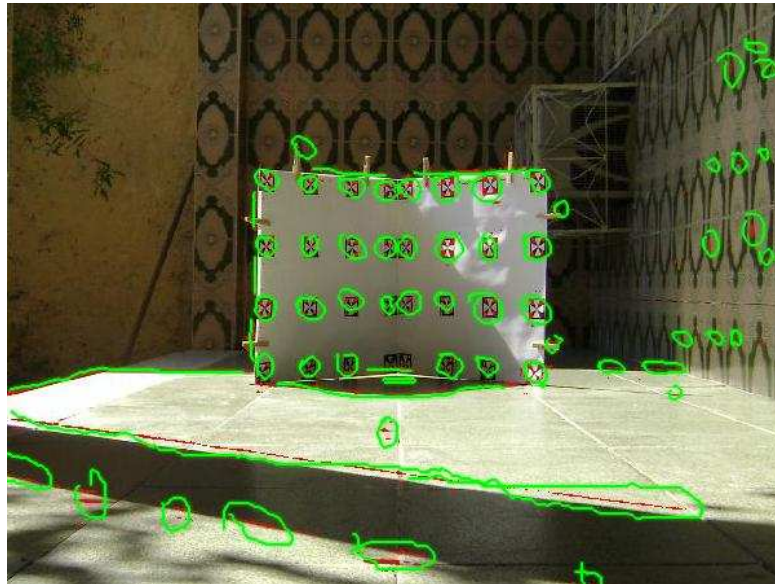


Figura 42 - Alvos identificados pela redeVP com grade a 2 metros

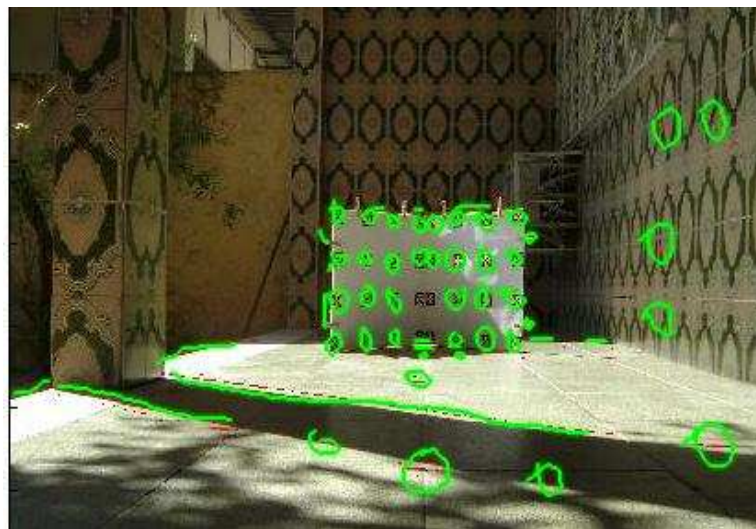


Figura 43 - Alvos identificados pela redeVP com grade a 3 metros

Pela soma de R_1 , R_2 e R_3 , tem-se $\Omega = 1,78$.

Alvo tipo X

Neste item serão apresentados os resultados da varredura das imagens com o alvo do tipo X pela redeXP. Com a grade a 1 metro da câmera a rede obteve bons resultados, selecionando poucos falsos alvos. Contudo, com a grade a 2 e 3 metros, as figuras demonstram que a rede não conseguiu diferenciar com eficiência os alvos dos não alvos.

A figura 44 mostra os alvos com a grade a 1 metro de distância da câmera e três falsos alvos foram selecionados, apresentando $R1 = 0,91$.



Figura 44 - Alvos identificados pela redeXP com grade a 1 metro

As Figuras 45 e 46 demonstram o péssimo desempenho alcançado pela redeXP com a grade posicionada a 2 e 3 metros da câmera, respectivamente. Além de todos os alvos, a rede identificou diversos falsos alvos, apresentando $R2 = 0,41$ e $R3 = 0,47$.

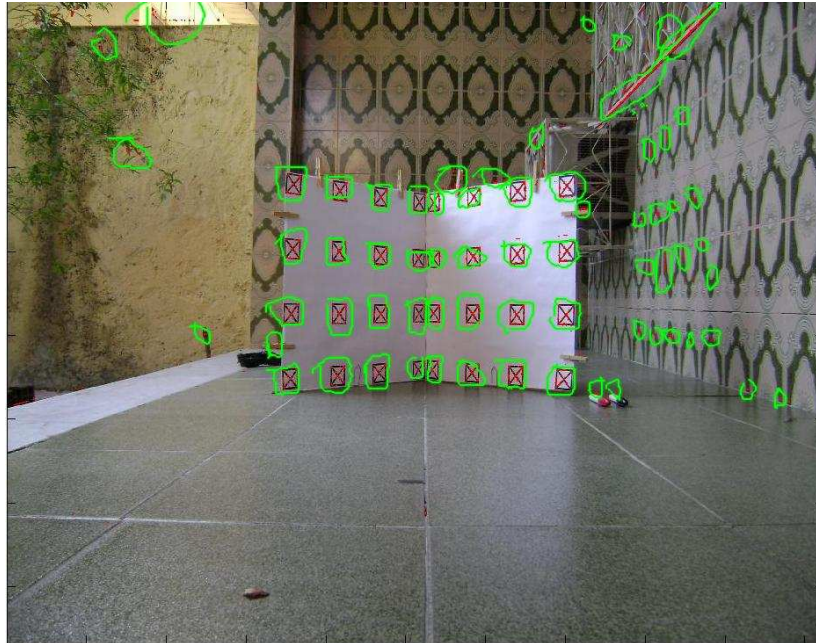


Figura 45 - Alvos identificados pela redeXP com grade a 2 metros

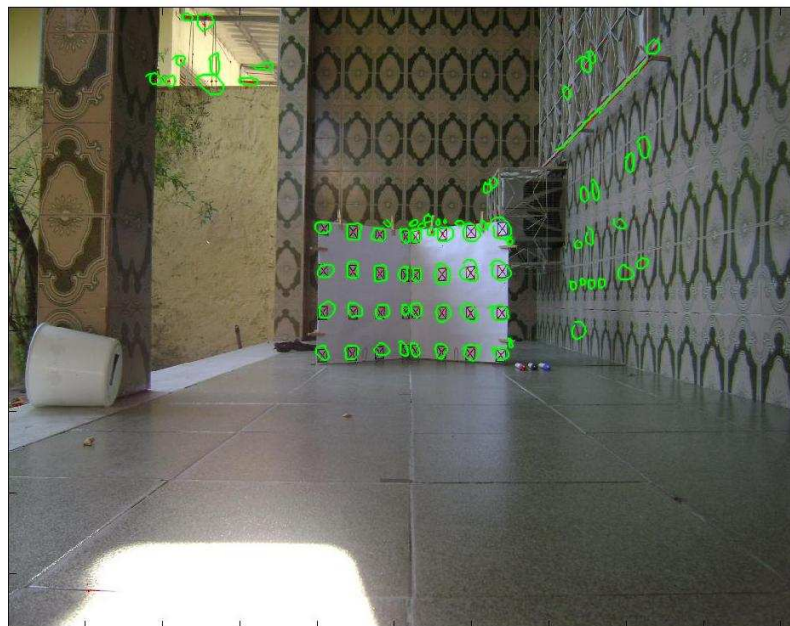


Figura 46 - Alvos identificados pela redeXP com grade a 3 metros

Pela soma de R1, R2 e R3, obtêm-se $\Omega = 1,79$.

Na Tabela 1 é apresentado o Ω para cada tipo de alvo.

Tipos de Alvo	Ω
Alvo tipo círculo	2,79
Alvo tipo X	1,79
Alvo tipo vasco	1,78

Tabela 1- Tipos de alvo x Ω

4.1.2 Teste 2

A partir dos resultados observados no primeiro teste, elaborou-se o teste 2, o qual consistiu em utilizar a redeCP, já que esta apresentou o maior $\Omega = 2,79$ para processar imagens de uma nova grade contendo 24 alvos, inserida em um novo cenário.

Para reduzir os falsos alvos identificados, as duas colunas centrais de alvos foram eliminadas e antes de a imagem ser processada, esta deve ter apenas a grade selecionada, eliminando o cenário em volta da mesma, como pode ser observado na Figura 47.

Este procedimento não inviabiliza o uso da rede neural para a identificação, pois é um processo que o operador, ao usar o sistema de calibração de imagens, pode realizar definindo, com o *mouse*, um retângulo ao redor da grade.



Figura 47 – Grade posicionado a 2 metros da câmera

A nuvem de pixel pode ser eliminada pela implementação de uma rotina no Matlab, denominada “clusteriza”, a qual separa por classes as nuvens de pixels, relacionando suas coordenadas x e y com o número total de alvos. O código desta função pode ser observado no Apêndice B.

A idéia é agrupar os pontos identificados como alvo e separá-los em classes de acordo com sua proximidade, uma classe para cada alvo, e então realizar a média e montar o vetor dos alvos reais, para saber se está correto é gerada também o vetor para pintar os alvos na imagem.

A rotina pode ser formalizada por:

1. Entrar com a matriz de alvos, gerada pela rede, o tamanho da imagem e o nome da imagem;
2. Reduzir a quantidade de dados da matriz, deixando um total de 2 vezes a quantidade de alvos presente na grade, gerando a matriz “alvo novo”;
 - Usando a função do Matlab *Kmeans*, a matriz de alvos é reduzida para um total de 48 alvos.
 - Esta função reduz a matriz de entrada para a quantidade desejada.
3. Ordenar os pontos na direção do eixo X, agrupando os pontos em colunas, o mesmo número de colunas de alvos da grade;
 - Realiza-se a diferença de x com o próximo x. Todos os pontos que tiverem uma diferença menor que,

$$\text{diferença} = \frac{\frac{\text{largura da imagem}}{\text{número de colunas} - 1}}{2}$$

- estão na mesma coluna. Quando a diferença é maior que a expressão apresentada, troca de coluna.
4. Para cada coluna ordenar os pontos na direção do eixo y, agrupando os pontos, pelo mesmo número de alvos da coluna;
 - Realiza-se a diferença de y com o próximo y. Todos os pontos que tiverem uma diferença menor que,

$\text{diferença} = \frac{\frac{\text{altura da imagem}}{\text{número de linhas} - 1}}{2}$
--
- estão no mesmo alvo. Quando a diferença é maior, troca de alvo.
5. Montar a matriz “alvo certo”, fazendo a média de cada elemento da matriz “alvos”
 - Matriz “alvo certo” é a matriz formada pela média de cada alvo, é uma matriz com 4 linhas e 6 colunas.
 - Cada elemento da matriz alvo certo pode ser associado a uma coordenada x, y, z da grade
 6. Montar a matriz “alvos para ver”, a partir da matriz “alvo certo”;
 - Matriz “alvos para ver” é uma lista de pixels identificados como alvo.

Na Figura 48 temos os alvos identificados com a grade à 1 metro da câmera. Para uma melhor observação, os cinco pixels no entorno do alvo identificado foram pintados de vermelho. Apenas dois alvos não foram identificados corretamente.



Figura 48 - Grade posicionada a 1 metro da câmera

No detalhe, a figura 49 mostra apenas o pixel “alvo” pintado de vermelho. Como pode ser observado, o pixel central do alvo foi selecionado com precisão.

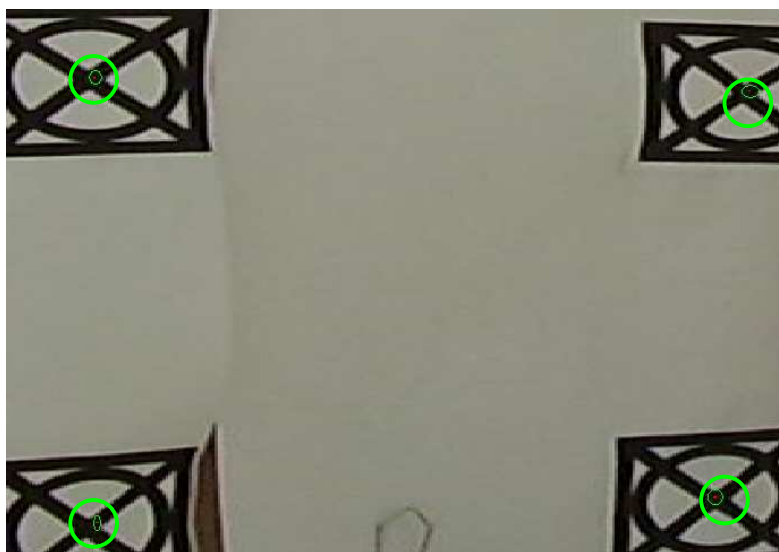


Figura 49 - Zoom da grade a 1 metro da câmera

A rede mostrou-se eficiente também com as grades à 2 e 3 metros da câmera, como pode ser observados nos resultados apresentados nas figuras 50 e 51.



Figura 50 - Grade a 2 metros da câmera



Figura 51 - Grade a 3 metros da câmera

As Figuras 52 e 53 mostram que mesmo com a grade virada para a direita ou para a esquerda, isto é, alguns dos alvos vistos de lado, a rede foi capaz de identificá-los com boa precisão, apresentando poucos erros.



Figura 52 - Grade a 1 metro da câmera virada para a direita



Figura 53 - Grade a 1 metro da câmera virada para a esquerda

Os resultados apresentados demonstram que a redeCP identificou de maneira eficaz o alvo do tipo círculo nas três distâncias apresentadas. Tais dados poderiam ser inseridos em uma rotina de calibração, permitindo a execução automática deste processo.

Capítulo 5

Conclusão

O objetivo principal do projeto era investigar alguns padrões de alvos e avaliar qual padrão apresentaria os maiores índices de acerto para uma rede neural. Essa rede, então deveria ser capaz de identificar os alvos em uma imagem desconhecida e, através de pós-processamento de dados, associar os pixels centrais dos alvos a pontos (x,y,z) do espaço real. Na etapa de pré-processamento (seleção dos alvos) das redes, buscou-se cobrir o máximo possível de possibilidades de posições e distâncias dos alvos, para que esta aplicação pudesse ser utilizada em diversos processos de reconhecimento de padrões em imagens. Para que a rede funcionasse corretamente, foi empregado o treinamento através do algoritmo Backpropagation. Para realizar os testes, foram adotadas imagens de grades com três diferentes tipos de alvos. A partir dessas imagens, as informações das coordenadas horizontais e verticais dos alvos deviam ser determinadas.

No primeiro teste realizado, todas as etapas descritas na metodologia foram executadas. O resultado final foi a localização de “nuvens” de pixels em torno dos pixels centrais dos elementos alvo. Além disso, as duas colunas centrais de alvos geraram erroneamente elementos reconhecidos pela rede, fazendo com que fosse necessária a implantação de técnicas de pós-processamento. Neste teste também identificou-se que o tipo de alvo mais indicado para a calibração seria o do tipo “Círculo”.

A partir dos resultados do teste 1, foi vista a necessidade da implantação de mecanismos que atuassem na redução da dispersão dos resultados obtidos. A eliminação das duas

colunas centrais de alvos eliminou o problema de a rede identificar os dois alvos centrais como sendo apenas um e a seleção apenas da grade antes do processamento da imagem pela rede reduziu em muito a quantidade de falsos alvos identificados pela rede. Contudo, o resultado final emitido pela rede continuava nos apresentando conjuntos de coordenadas de pixels em torno dos alvos das imagens processadas. Como o objetivo é determinar a coordenada do pixel central de um alvo, foi desenvolvido um algoritmo de clusterização, que se baseou no agrupamento (através de médias) de coordenadas de pixels “próximos”, associando ao número de alvos na grade. A rotina também tem como saída a correspondência da lista de alvos identificados com a posição espacial real do alvo na grade de calibração.

O teste 2 utilizou uma nova grade com 24 alvos do tipo círculo e restrição do processamento apenas da imagem com a grade. O teste apresentou apenas um erro quando a grade estava posicionada a 3 metros da câmera (Figura 51), nenhum erro com a grade a 2 metros (Figura 50) e um erro com a grade a 1 metro (Figura 48). A partir dos resultados pode-se concluir esta configuração utilizada da grade pode ser considerada “ótima”. E pela aplicação do pós-processamento o objetivo de reconhecimento dos pixels centrais dos alvos foi alcançado.

Pela análise de todos os resultados apresentados foi concluído que este processo realmente pode ser adotado na calibração de câmeras em estereofotogrametria, o que automatizaria uma etapa trabalhosa (determinação das coordenadas dos alvos, normalmente realizada manualmente) e também fruto de muitos erros. Além disso, este processo pode ser utilizado em muitas outras aplicações na área da eletrônica, reduzindo

custos de projeto, tanto em termos de mão-de-obra quanto em termos tecnológicos, já que este se mostrou bastante eficiente, utilizando poucos recursos computacionais.

Para trabalhos futuros, sugere-se aumentar a quantidade de dados de entrada e implementar uma rotina que seja capaz de reconhecer com eficiência a grade em diferentes posições. Deve ser implementada também uma rede capaz de identificar diferentes tipos de alvo na mesma imagem. O tamanho da janela deve ser aumentado, para que englobe melhor a imagem com a grade a 1 metro da câmera. Sugere-se também o uso de redes especializadas para cada distância, pois os resultados aparentemente são melhores para alvos treinados apenas para essa distância.

Referências Bibliográficas

- [1] Haykin, Simon, 1931 – Neural networks: a comprehensive foundation – 2nd ed – Pretintice-Hall, 1999.
- [2] Hert, John – Introduction to the Theory of Neural Computation – Lecture Notes Volume I – Santa Fe Institute Editorial Boar, 1990.
- [3] Silva, L.C. – Método Robusto para Calibração de Câmeras em Estereofotogrametria – Tese, COPPE-UFRJ, 2003.
- [4] Acchar, Juliana – Reconhecimento de padrões de calibração em estereofotogrametria através de rede neurais – Projeto de final de Graduação – DEL, UFRJ, 2008.
- [5] Tutorial de Redes Neurais - Aplicações em Bioinformática
http://www.lncc.br/~labinfo/tutorialRN/frn4_backpropagation.htm - em 03/02/2009
- [6] JAIN, A.K. – Data Clustering: A Review - ACM Computing Surveys, Vol. 31, No. 3, September 1999. - <http://www.cs.rutgers.edu/~mlittman/courses/lightai03/jain99data.pdf> - em 03/02/2009

Apêndice A

Treinamento das Redes

Alvo do tipo Círculo

Grade a 1 metro da câmera

Esta rede foi nomeada de “redeCP1m”.

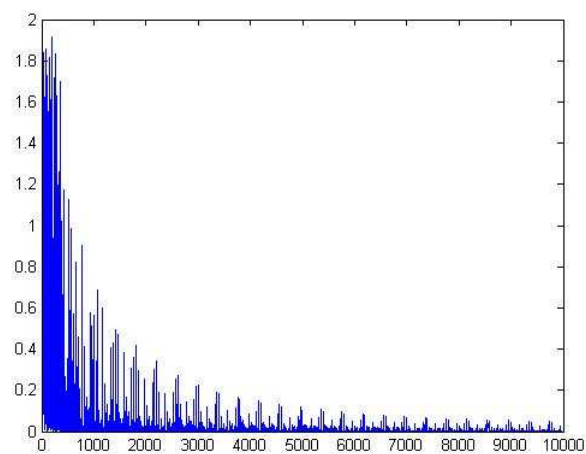


Figura 54 - Curva de aprendizagem da redeCP1m

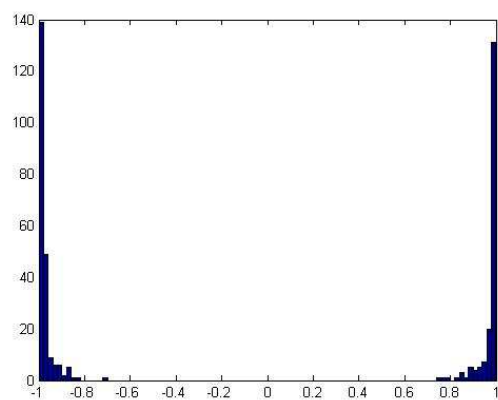


Figura 55 - Histograma do treino da redeCP1m.

Grade a 2 metros da câmera

Esta rede foi nomeada de “redeCP2m”.

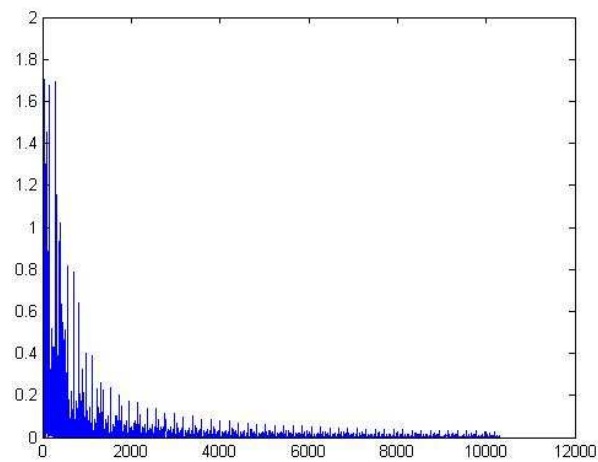


Figura 56 - Curva de aprendizagem da redeCP2m.

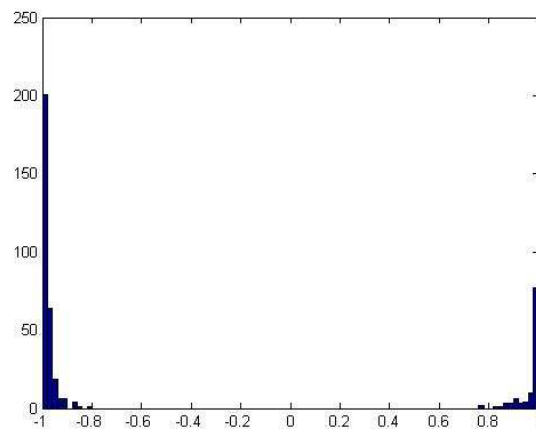


Figura 57 - Histograma do treinamento da redeCP2m.

Grade a 3 metros da câmera

Esta rede foi nomeada de “redeCP3m”.

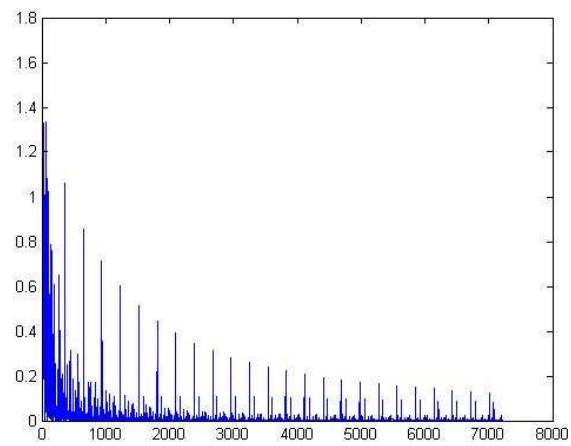


Figura 58 - Curva de aprendizagem da redeCP3m.

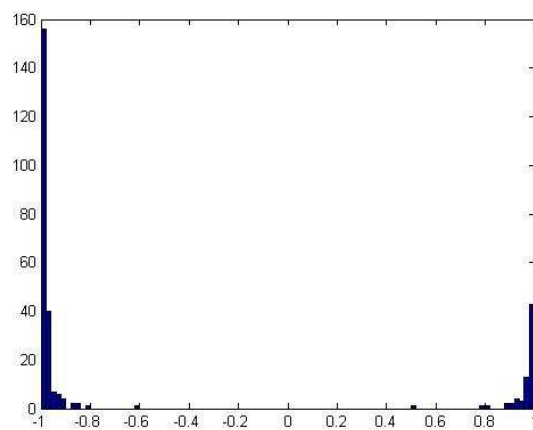


Figura 59 - Histograma do treinamento da redeCP3m

Rede única

Esta rede foi nomeada de “redeCP”.

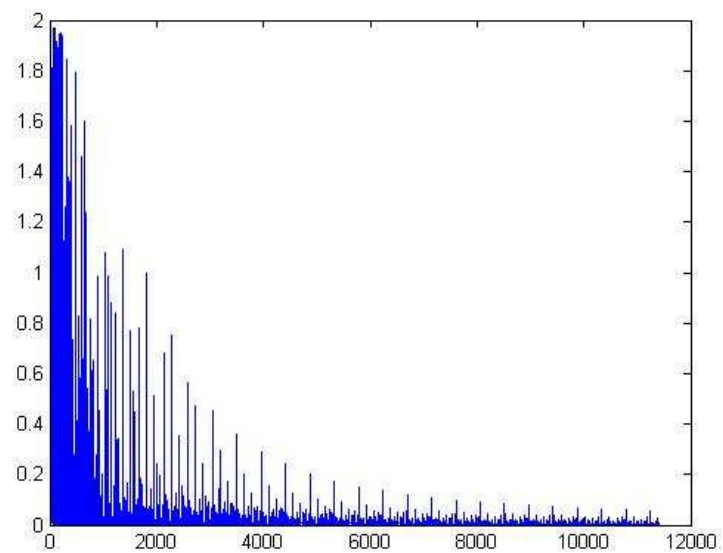


Figura 60 - Curva de aprendizagem da redeCP

O gráfico acima demonstra que a rede conseguiu ser bem treinada para as três distâncias do grade.

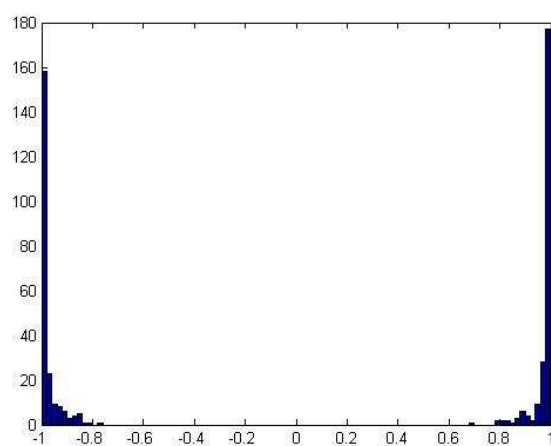


Figura 61 - Histograma do treinamento da redeCP

Alvo do tipo Vasco

Grade a 1 metro da câmera

Esta rede foi nomeada de “redeVP1m”.

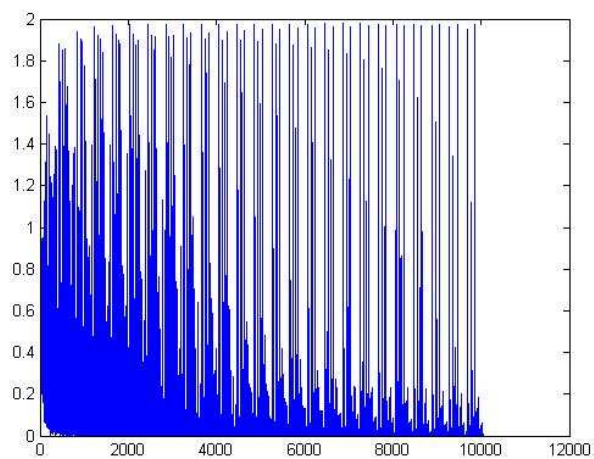


Figura 62 - Curva de aprendizagem da redeVP1m

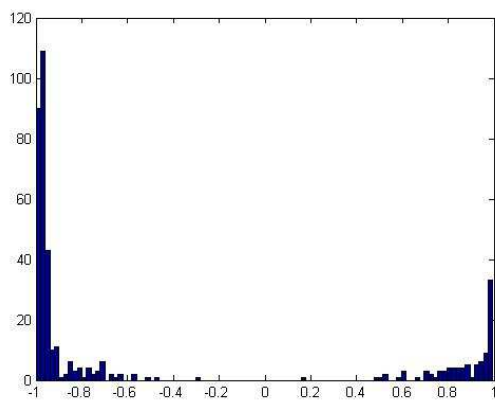


Figura 63 - Histograma do treinamento da redeVP1m

Grade a 2 metros da câmera

Os gráficos abaixo foram obtidos a partir do treinamento de uma rede tendo como entrada os alvos e não alvos do tipo Vasco a uma distância de 2 metros da câmera. Esta rede foi nomeada de “redeVP2m”.

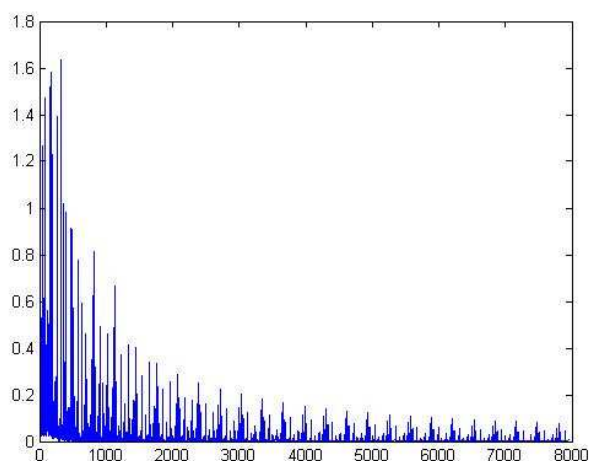


Figura 64 - Curva de treinamento da redeVP2m

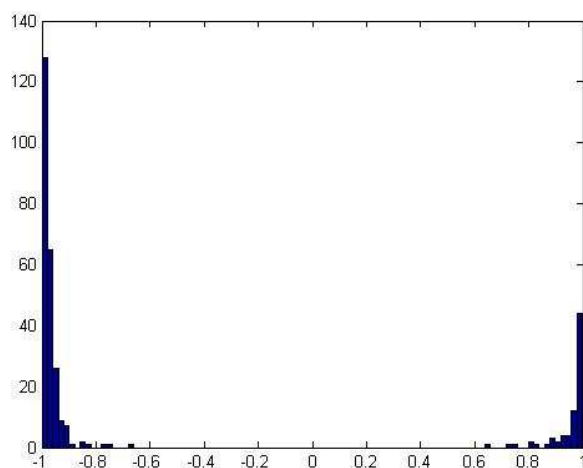


Figura 65 - Histograma do treinamento da redeVP2m

Grade a 3 metros da câmera

Esta rede foi nomeada de “redeVP3m”.

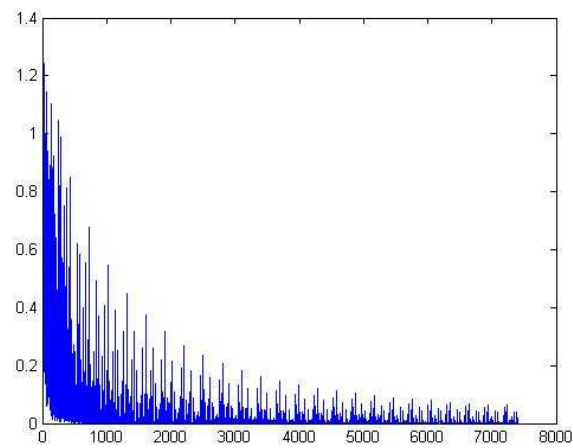


Figura 66 - Curva de treinamento da redeVP3m

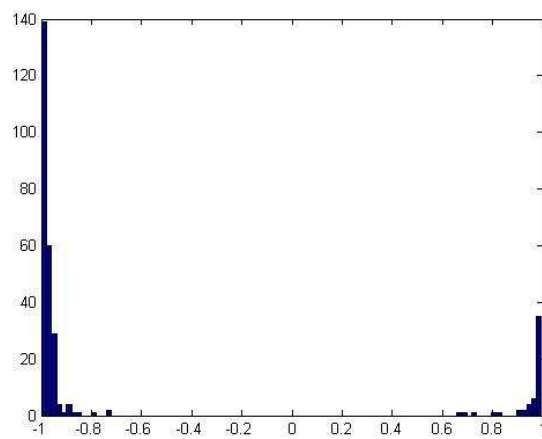


Figura 67 - Histograma do treinamento da redeVP3m

Rede Única

Esta rede foi nomeada de “redeVP”.

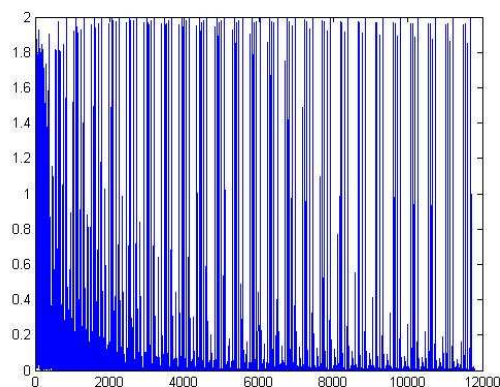


Figura 68 - Curva de treino da redeVP

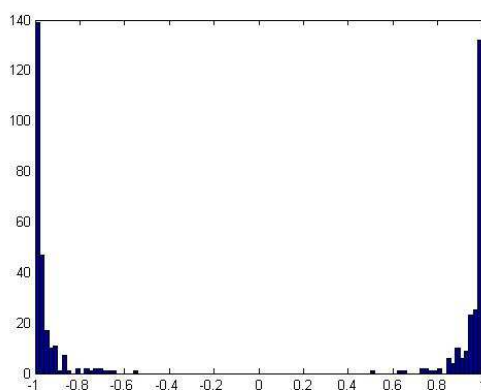


Figura 69 - Histograma do treinamento da redeVP

Os gráficos acima demonstram o que a redeVP não obteve um bom treinamento para as três distâncias do alvo, tal resultado deve-se ao fato dos alvos localizados a 1 metro da câmera não terem sido eficaz no treinamento da redeVP1m. Para comprovar este fato,

uma nova rede foi treinada, excluindo-se os alvos localizados a 1 metro da câmera. Esta rede foi chamada de redeVP2e3m e seus gráficos podem ser observados abaixo.

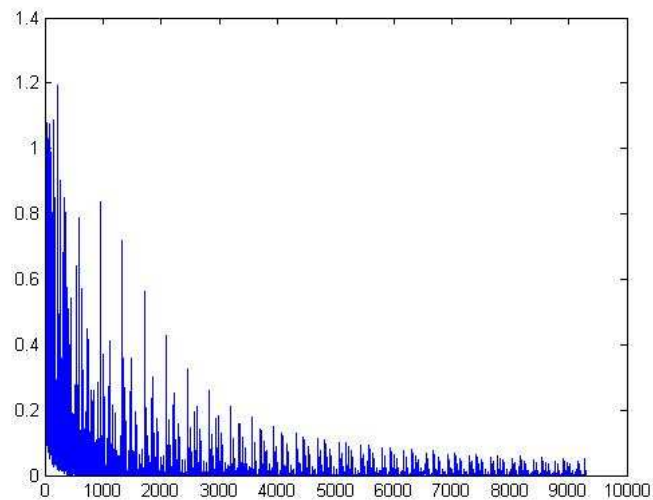


Figura 70 - Curva de treinamento da redeVP2e3m

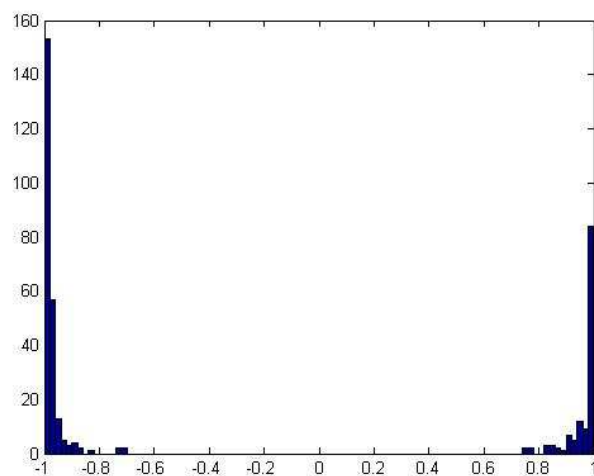


Figura 71 - Histograma de treinamento da redeVP2e3m

De acordo com os gráficos a rede obteve um bom treinamento e um limiar de 0,7. Comprovando que o problema encontra-se no conjunto de dados a 1 metro de distância da câmera.

Alvo do tipo X

Grade a 1 metro da câmera

Esta rede foi nomeada de “redeXP1m”.

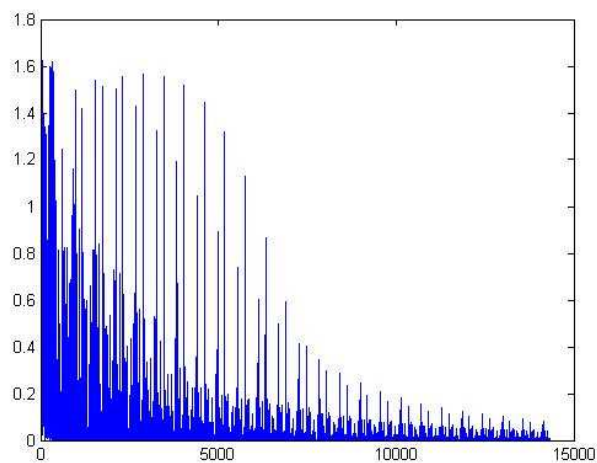


Figura 72 - Curva de treinamento da redeXP1m

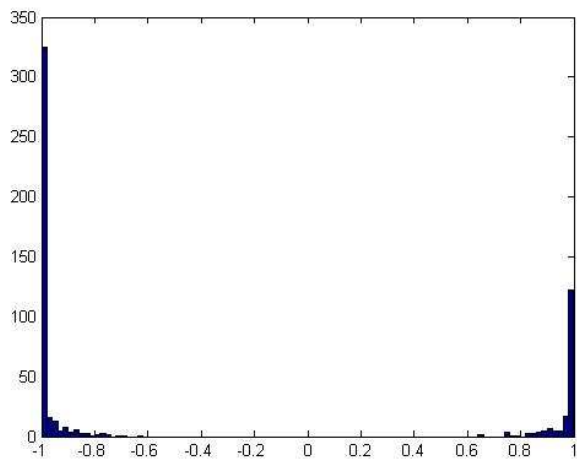


Figura 73 - Histograma do treino da redeXP1m

Grade a 2 metros da câmera

Os gráficos abaixo foram obtidos a partir do treinamento de uma rede tendo como entrada os alvos e não alvos do tipo X a uma distância de 2 metros da câmera. Esta rede foi nomeada de “redeXP2m”.

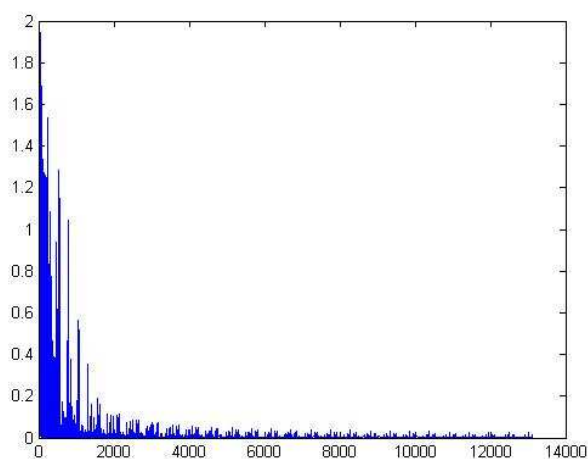


Figura 74 - Curva de treino da redeXP2m

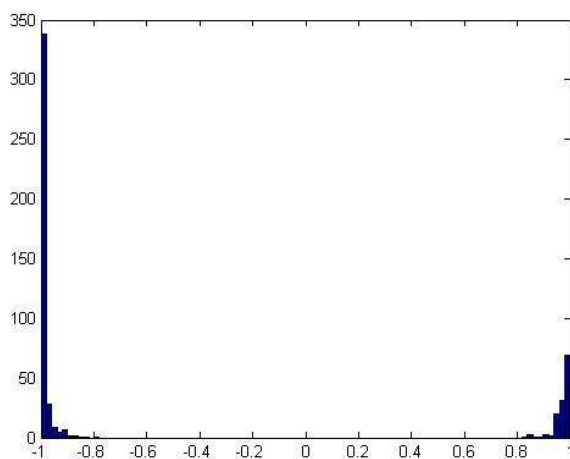


Figura 75 - Histograma do treino da redeXP2m

Grade a 3 metros da câmera

Os gráficos abaixo foram obtidos a partir do treinamento de uma rede tendo como entrada os alvos e não alvos do tipo X a uma distância de 3 metros da câmera. Esta rede foi nomeada de “redeXP3m”.

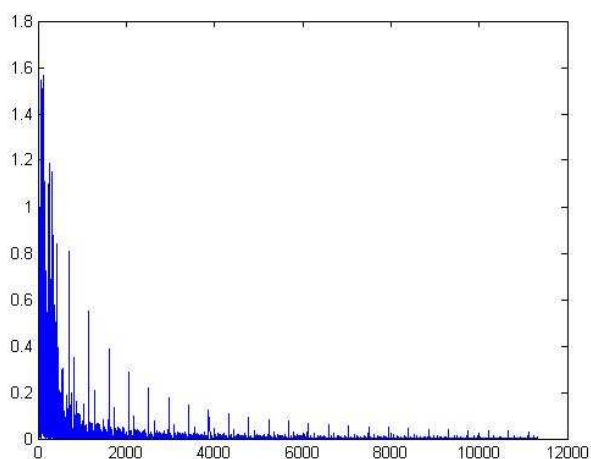


Figura 76 - Curva de treinamento da redeXP3m

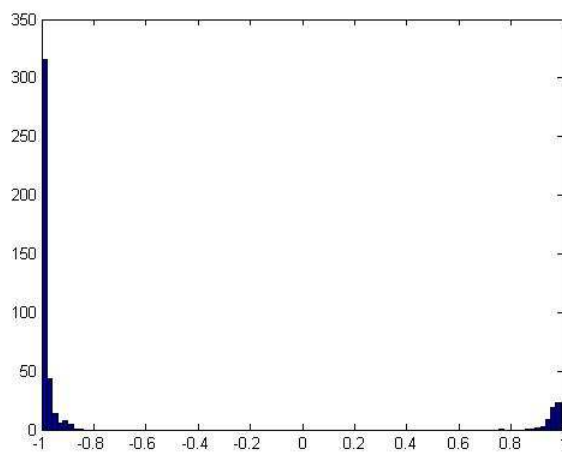


Figura 77 - Histograma de treino da redeXP3m

Rede Única

Os gráficos abaixo foram obtidos a partir do treinamento de uma rede tendo como entrada os alvos e não alvos do tipo X nas três distâncias da câmera. Esta rede foi nomeada de “redeXP”.

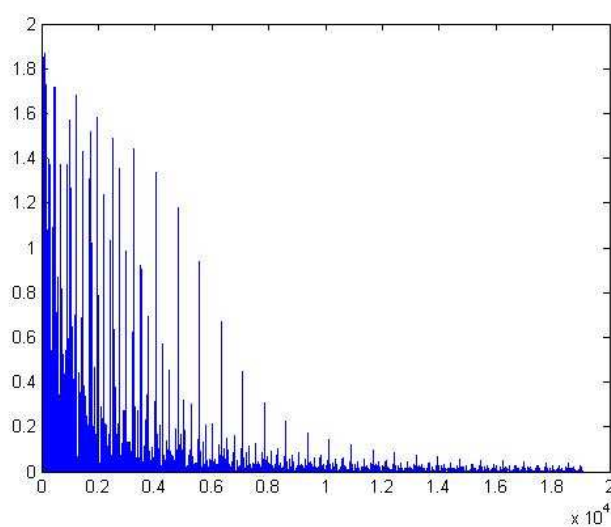


Figura 78 - Curva de treinamento da redeXP

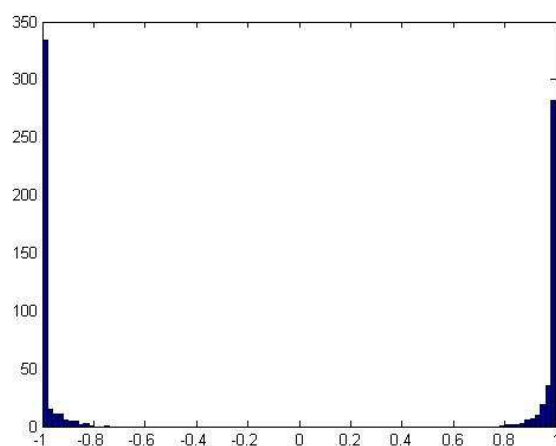


Figura 79 - Histograma de treino da redeXP

Teste das redes

Alvo do tipo Círculo

Grade a 1 metro da câmera

Neste tópico será apresentado o teste realizado para a redeCP1m.

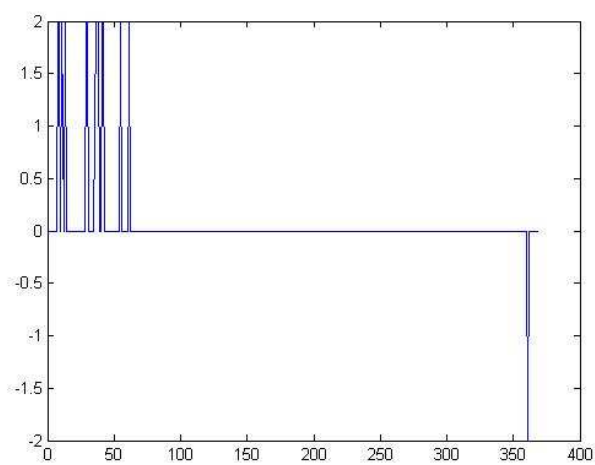


Figura 80 - Gráfico de erro referente ao teste da redeCP1m

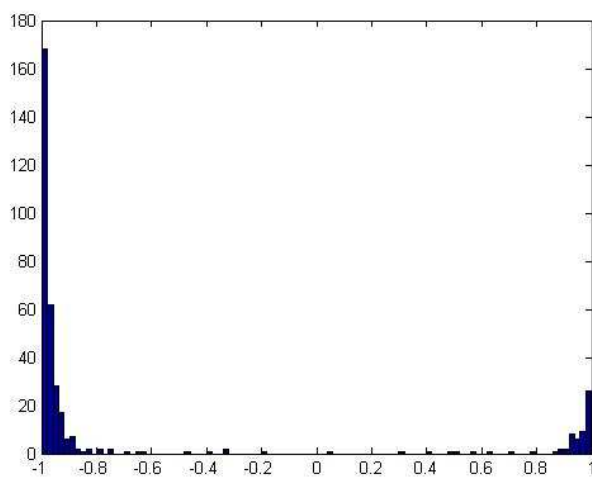


Figura 81 - Histograma do teste da redeCP1m

Grade a 2 metros da câmera

Neste tópico será apresentado o teste realizado para a redeCP2m.

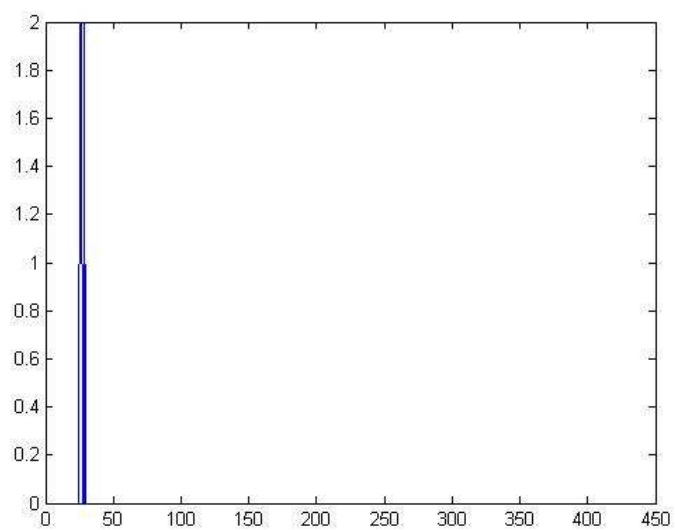


Figura 82 - Curva de erro do teste referente à redeCP2m

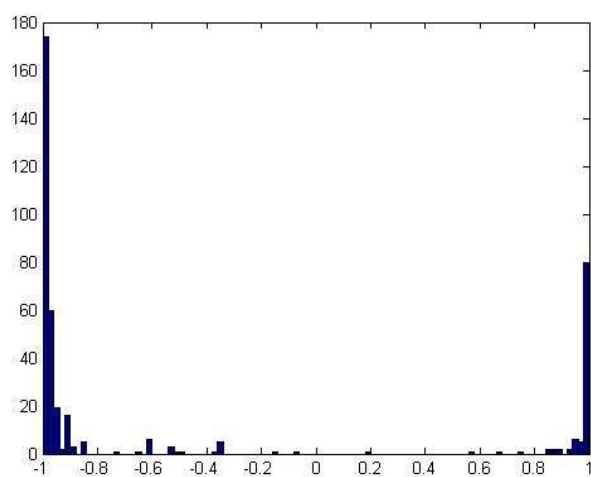


Figura 83 - Histograma do teste da redeCP2m

Grade a 3 metros da câmera

Neste tópico será apresentado o teste realizado para a redeCP3m.

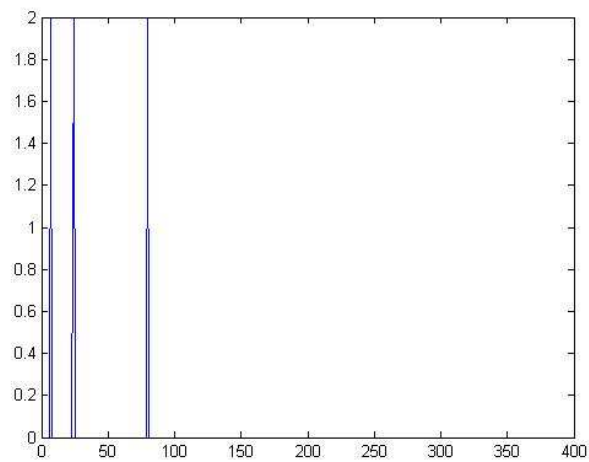


Figura 84 - Curva de erro referente ao teste da redeCP3m

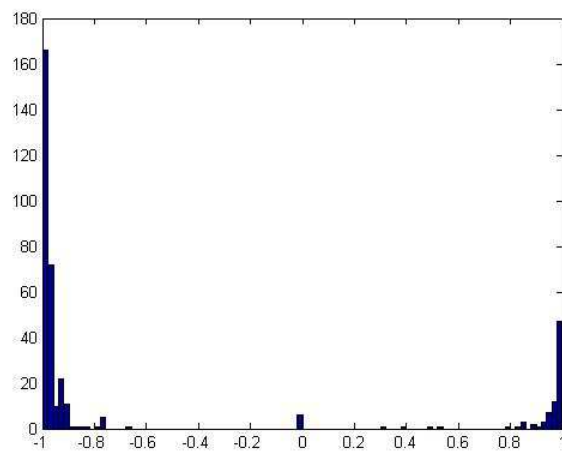


Figura 85 - Histograma do teste da redeCP3m

Rede Única

Neste tópico será apresentado o teste realizado para a redeCP.

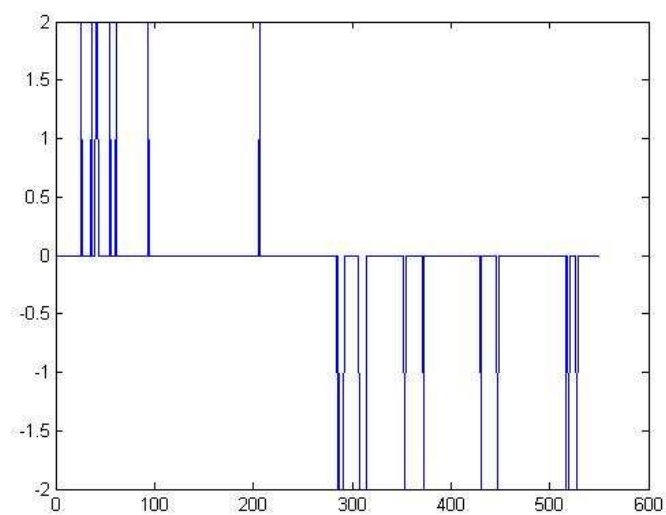


Figura 86 - Curva de erro referente ao teste da redeCP

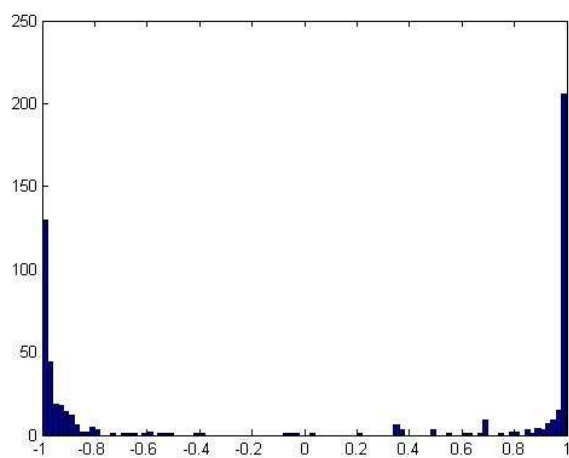


Figura 87 - Histograma do teste da redeCP

Alvo do tipo Vasco

Grade a 1 metro da câmera

Neste tópico será apresentado o teste realizado para a redeVP1m.

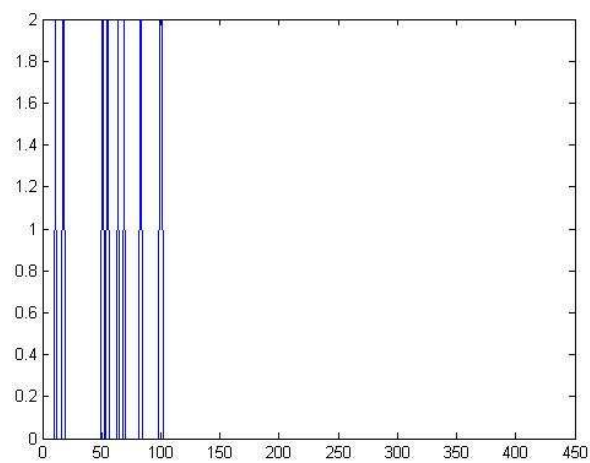


Figura 88 - Curva de erro referente à redeVP1m

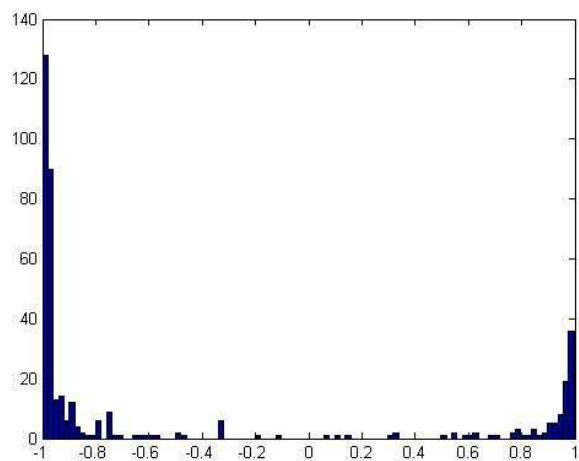


Figura 89 - Histograma do teste da redeVP1m

Grade a 2 metros da câmera

Neste tópico será apresentado o teste realizado para a redeVP2m.

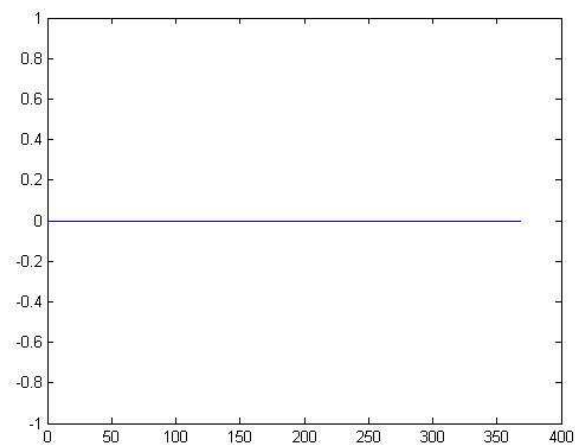


Figura 90 - Curva de erro referente à redeVP2m

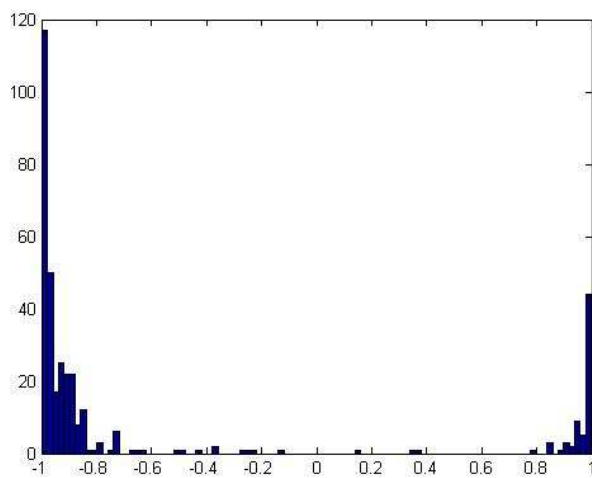


Figura 91 - Histograma do teste da redeVP2m

Grade a 3 metros da câmara

Neste tópico será apresentado o teste realizado para a redeVP3m.

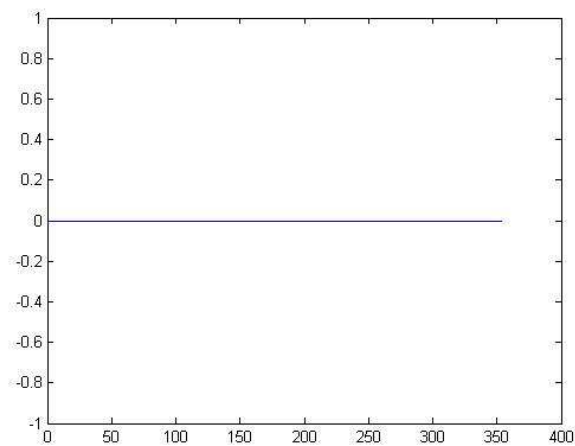


Figura 92 - Curva de erro referente à redeVP3m

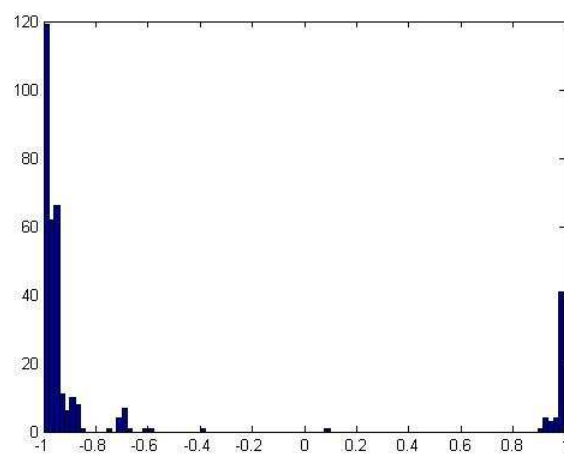


Figura 93 - Histograma do teste da redeVP3m

Rede Única

Neste tópico será apresentado o teste realizado para a redeVP.

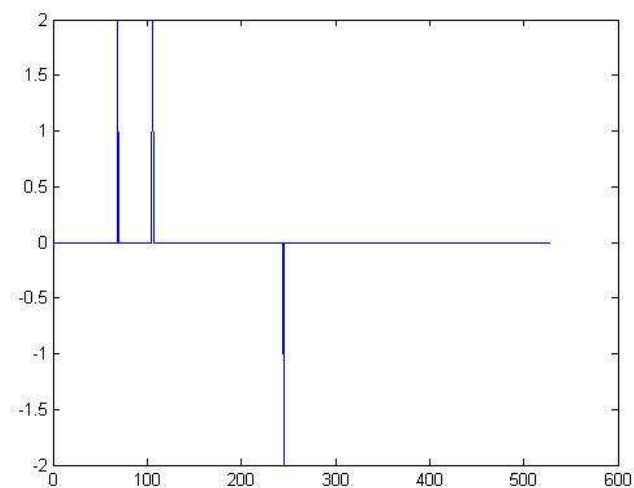


Figura 94 - Curva de erro referente à redeVP

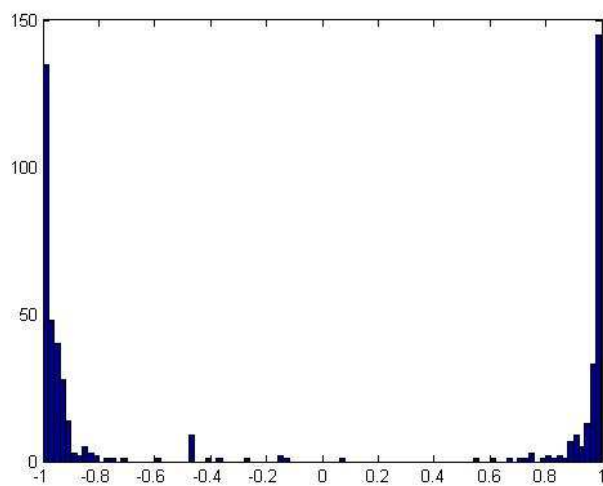


Figura 95 - Histograma do teste da redeVP

A seguir apresentaremos os resultados para o teste da redeVP2e3m, foram utilizados como dados de entrada alvos localizados a 2 e 3 metros da câmera.

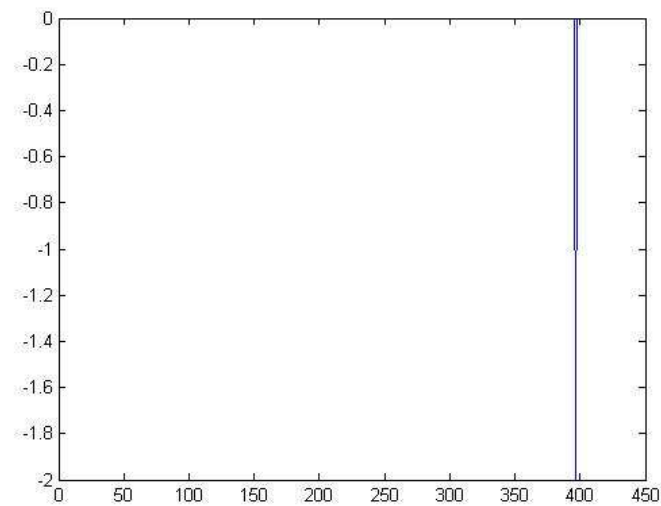


Figura 96 - Curva de erro referente à redeVP2e3m

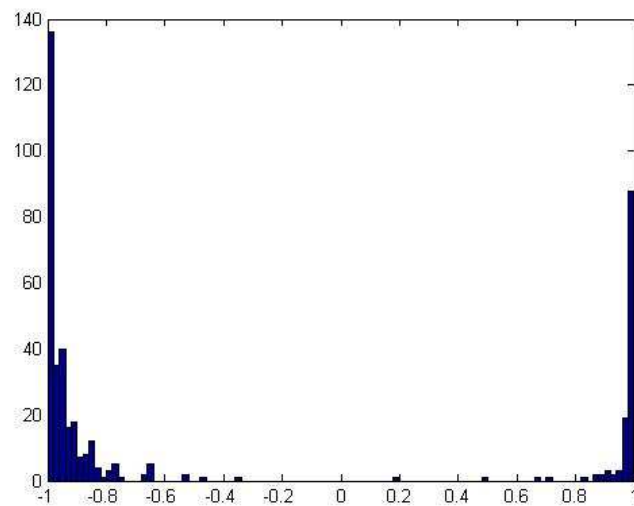


Figura 97 - Histograma do teste da redeVP2e3m

Alvo do tipo X

Grade a 1 metro da câmera

Neste tópico será apresentado o teste realizado para a redeXP1m.

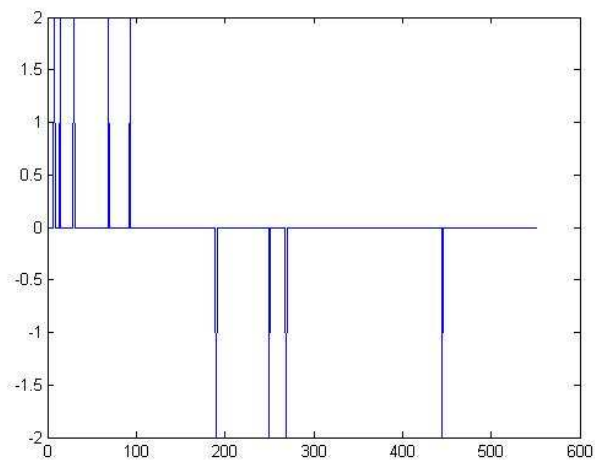


Figura 98 - Curva de erro do teste referente à redeXP1m

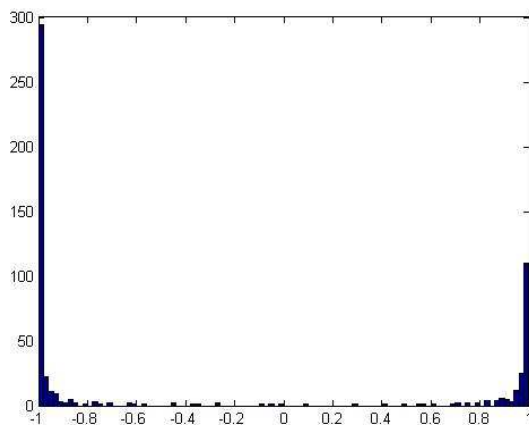


Figura 99 - Histograma do teste da redeXP1m

Grade a 2 metros da câmera

Neste tópico será apresentado o teste realizado para a redeXP2m.

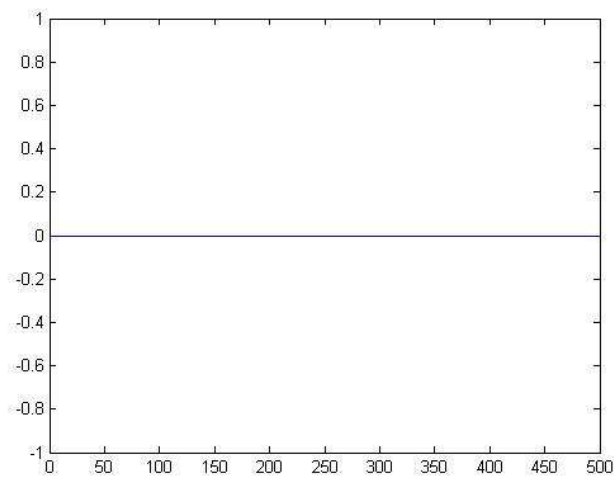


Figura 100 - Curva de erro do teste referente à redeXP2m

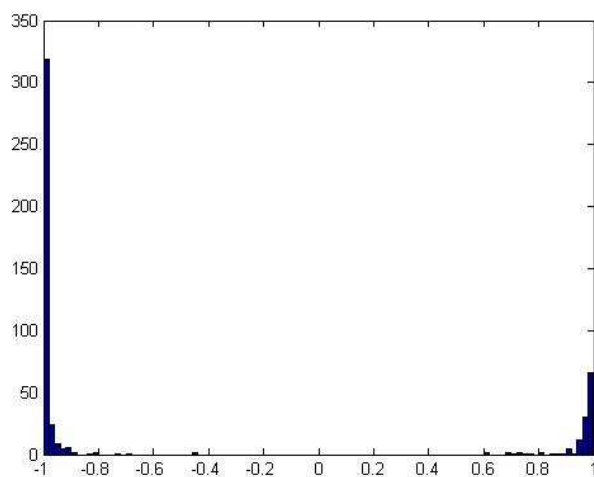


Figura 101 - Histograma do teste da redeXP2m

Grade a 3 metros da câmera

Neste tópico será apresentado o teste realizado para a redeXP3m.

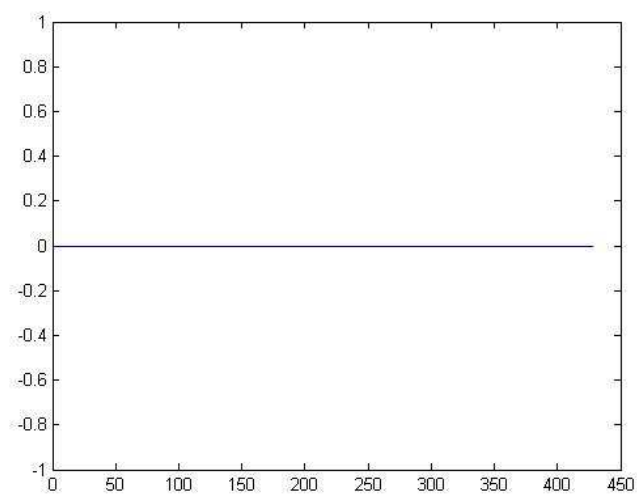


Figura 102 - Curva de erro do teste referente à redeXP3m

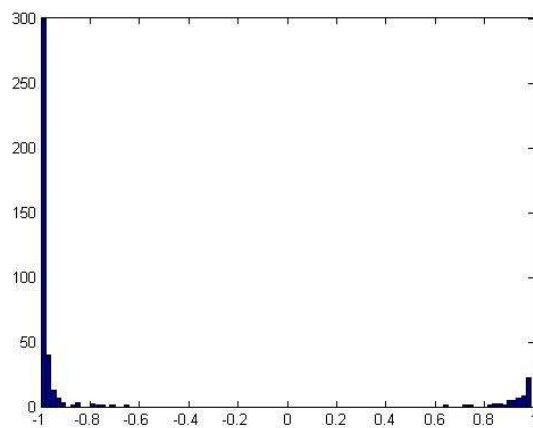


Figura 103 - Histograma do teste da redeXP3m

Rede Única

Neste tópico será apresentado o teste realizado para a redeXP.

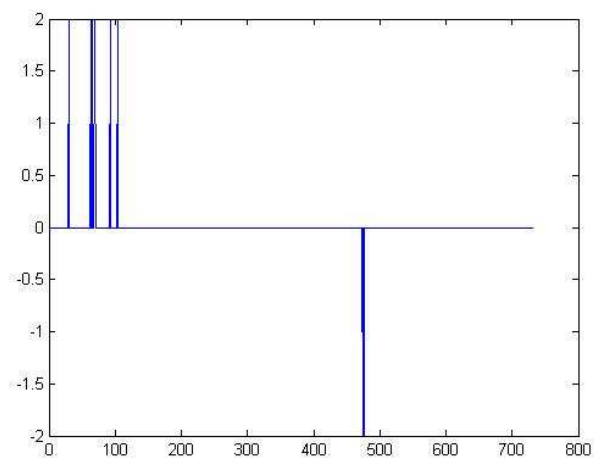


Figura 104 - Curva de erro do teste referente à redeXP

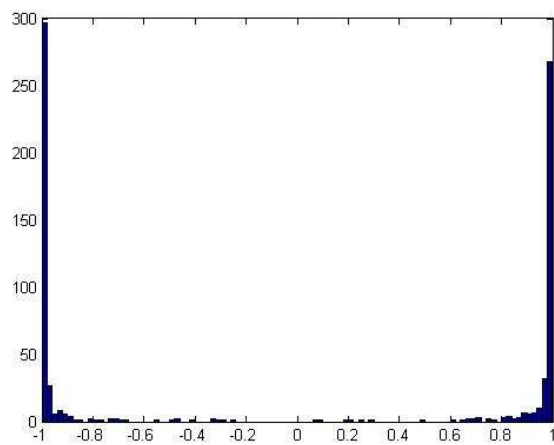


Figura 105 - Histograma do teste da redeXP

Apêndice B

Rotina Clusteriza

```
function [alvos_certos,alvos_para_ver]=clusteriza(C,Img_size,nome_da_imagem)

% [alvos_certos,alvos_para_ver]=clusteriza(C,Img_size,nome_da_imagem)

% onde:

% C é a lista de saída da rede.

% Img_size é um vetor com o tamanho da imagem: ex: Img_size = [ 1600 1200]

% alvos_certos é uma estrutura em forma matricial com os alvos encontrados

% alvos_para_ver é uma lista com os alvos certos apenas par avisualização

% nome_da_imagem é a imagem onde serão pintados os alvos

Width = Img_size(1);

Height = Img_size(2);

[i,centros] = kmeans(C,48);

centros = round(centros);

Y = sortrows(centros,1);

Ydif = diff(Y(:,1));

Gap=find (Ydif >= Width/5*.3) % largura/numero de espaços /2

Gap = [ 0 ; Gap ; length(Y)];

for n=1:length(Gap)-1

    coluna{n} = Y((Gap(n)+1):Gap(n+1),:)

end

for col=1:6
```

```

Y = sortrows(coluna{col},2);

Ydif = diff(Y(:,2));

Gap = find(Ydif >= Height/3*.3) % largura/numero de espaços /2

Gap = [ 0 ; Gap ; length(coluna{col})];

for n=1:length(Gap)-1

    alvos{col,n} = Y((Gap(n)+1):Gap(n+1),:)

end

end

%alvos médios

alvos_para_ver = [];

for i=1:size(alvos,1)

    for j=1:size(alvos,2)

        alvos_certos{i,j} = round(mean(alvos{i,j},1));

        alvos_para_ver = [ alvos_para_ver ; alvos_certos{i,j} ];

    end

end

alvos_certos = alvos_certos';

mostra_alvo(alvos_para_ver,nome_da_imagem)

```

Rotina Treina Rede

```
function [L,tfinal,erro] = treinarede(nome_arquivo_alvo,nome_arquivo_nao_alvo)

%load alvos

%load naltvos

load(nome_arquivo_alvo)

load(nome_arquivo_nao_alvo)

num_alvos = size(matrizAlvos);

num_nao_alvos = size(matrizNaoAlvos);

% [A] Data

% substituir pelos seus dados (X e o t):

NUM_DADOS = num_alvos(1) + num_nao_alvos(1);

rand('state',0)

Xorig = [matrizAlvos ; matrizNaoAlvos]';

Media = mean(Xorig,2);

Xorig = Xorig - repmat(Media,1,NUM_DADOS);

% Extração das componentes principais - quantas?

R=Xorig*Xorig'/NUM_DADOS;

[V,D] = eig(R);

V = fliplr(V);

D = flipud(diag(D));

energ = cumsum(D)/sum(D);

i = find(energ >= 0.9);

N_comp = i(1);
```

```

for n=1:N_comp

    V90(:,n) = V(:,n)/norm(V(:,n));

end

X = V90'*Xorig;

% projeção de X no novo espaço

Input_size = N_comp;

% Normalizacao

STD = std(X,[],2);

%Saída desejada da rede (target)

%ex: t = [ 1 1 1 1 1 ... 1 -1 -1 -1 ... -1]

torig = [ones(1,num_alvos(1)) -ones(1,num_nao_alvos(1))];

% [A1] Randomize Data Order

% continua assim

randn('state',sum(100*clock))

%randn('state',0);

aux = [X ; torig ; randn(1,size(X,2))];

aux = sortrows(aux,size(aux,2));

X = aux(1:Input_size,:);

t = aux(Input_size+1,:);

clear aux;

% [B] Network Init

% [B1] Parameters

% continua assim

```

```

K = 2; % Number of Layers

L(1).K = K;

eta = 1e-2; % Learning Rate Initial Value

Delta = 1e-6; % Stop Criterion

N = size(X,2); % Number of Input Vectors

P = 25;

alpha = 1; % Learning Rate Decay Factor

mu = 0; % Momentum constant

% [B2] Layers

% inicializacao dos pesos da rede

for k=1:K-1

    L(k).W = rand(Input_size,Input_size)-0.5;

    L(k).b = rand(Input_size,1)-0.5;

end

L(K).W = rand(1,Input_size)-0.5;

L(K).b = rand(1,1)-0.5;

L(1).STD = STD;

%L(1).Vect = Vecs(:,1:N_comp);

L(1).Vect = V90;

L(1).Mean = Media;

%L (K) = K;

for k=1:K,

    L(k).vb = zeros(size(L(k).b));

```

```

    L(k).vW = zeros(size(L(k).W));

end;

% [C] Sequential Error Backpropagation Training

hh=0;

kk=0;

J=0;

Jtotal = 0;

erro = [];

for p=1:P

    for n=1:N

        % [C1] Feed-Forward

        L(1).x = X(:,n)./L(1).STD;

        for k = 1:K,

            L(k).u = L(k).W*L(k).x + L(k).b;

            L(k).o = tanh(L(k).u);

            L(k+1).x = L(k).o;

        end;

        e = t(n) - L(K).o;

        % [t(n) L(K).o e]

        erro = [ erro (e'*e)/2];

        % [C2] Error Backpropagation

        L(K+1).alpha = e;

```

```

L(K+1).W = eye(length(e));

for k = fliplr(1:K),

    L(k).M = eye(length(L(k).o)) - diag(L(k).o)^2;

    L(k).alpha = L(k).M*L(k+1).W'*L(k+1).alpha;

    L(k).db = L(k).alpha;

    L(k).dW = kron(L(k).x',L(k).alpha);

end;

% [C3] Updates

for k = 1:K,

    L(k).vb = eta*L(k).db + mu*L(k).vb;

    L(k).b = L(k).b + L(k).vb;

    L(k).vW = eta*L(k).dW + mu*L(k).vW;

    L(k).W = L(k).W + L(k).vW;

end;

% [C4] Stop criterion

eta = eta*alpha;

hh = hh+1;

kk = kk+1;

if(hh> 100)

    hh = 0;

    kk/(N*P)

end

end;

```



```

end

% [D] Test - Processamento pela rede

plot(erro);

certo = 0;

for n = 1:size(X,2),

    L(1).x = (L(1).Vect'*Xorig(:,n))./L(1).STD;

    for k = 1:K,

        L(k).u = L(k).W*L(k).x + L(k).b;

        L(k).o = tanh(L(k).u);

        L(k+1).x = L(k).o;

    end;

    tfinal(n) = L(K).o;

    %apresentacao dos resultados

    I = reshape(Xorig(:,n),31,31);

end;

figure(2)

hist (tfinal,100)

%save (L, 'L');

```

Rotina Testa Rede

```
function [tfinal,alvo] = testarede(L,nome_arquivo_alvo,nome_arquivo_nao_alvo)

%load alvos

%load nalvos

load(nome_arquivo_alvo)

load(nome_arquivo_nao_alvo)

num_alvos = size(matrizAlvos);

num_nao_alvos = size(matrizNaoAlvos);

% [A] Data

% substituir pelos seus dados (X e o t):

NUM_DADOS = num_alvos(1) + num_nao_alvos(1);

rand('state',0)

Xorig = [matrizAlvos ; matrizNaoAlvos]';

Media = mean(Xorig,2);

Xorig = Xorig - repmat(Media,1,NUM_DADOS);

torig = [ones(1,num_alvos(1)) -ones(1,num_nao_alvos(1))];

% [D] Test - Processamento pela rede

K = L(1).K

certo = 0;

for n = 1:size(Xorig,2),

    L(1).x = (L(1).Vect'*Xorig(:,n))./L(1).STD;

    for k = 1: K%L(1).K,
```

```

    L(k).u = L(k).W*L(k).x + L(k).b;

    L(k).o = tanh(L(k).u);

    L(k+1).x = L(k).o;

end;

tfinal(n) = L(K).o;

if(tfinal(n) > .6) % é alvo

    alvo(n) = 1;

else

    alvo(n) = -1;

end

end;

figure(1)

plot(torig-alvo)

erro= sum((abs((torig-alvo)/2))/length(torig))*100

figure(2)

hist (tfinal,100)

```

Rotina Processa Imagem

```
function alvo = processa_imagem(nome_da_imagem,L)

count=0;

ehora=0;

Grid = strcat('D:/UFRJ/PF/final/fotos_novas/',nome_da_imagem,'.bmp');

% % carregar a imagem da Grid

I = imread(Grid(1,:));

imgsize = size(I,1)*size(I,2);

% converte p/ preto e branco

[w, h, p]=size(I); % funcao conversao_cinza

Idouble = double(I); %para fazer os calculos

for i=1:w

    for j=1:h

        Ipb(i,j)=0.299*Idouble(i,j,1)+0.587*Idouble(i,j,2)+0.114*Idouble(i,j,3);

    end

end % END funcao conversao_cinza

K=2;

lado=31; %tamanho da matriz de janelamento

n=1;

alvo=[ ];

for i= 1:(size(Ipb,1)-lado) %for i=1:(61- lado)
```

```

for j= 1:(size(Ipb,2)-lado) %for j=2:40

    Win=Ipb(i:i+(lado-1),j:j+(lado-1));

    vWin = monta_vetor(Win); % pega a imagem 31x31 uma a uma e monta em um
vetor de 1X961

    L(1).x = (L(1).Vect'*(vWin'-L(1).Mean))./L(1).STD; % analisa se é alvo

    for k = 1:K,

        L(k).u = L(k).W*L(k).x + L(k).b;

        L(k).o = tanh(L(k).u);

        L(k+1).x = L(k).o;

    end;

    tfinal = L(K).o;

    if (tfinal > 0.6) % se for alvo

        alvo = [alvo;j+(lado-1)/2 i+(lado-1)/2]; %troca o i por j pois havia saído
transposto o resultado

        size(alvo);

    else

        nalvo = 0;

    end

    perc = count/imgsize*100;

    count=count+1;

    ehora=ehora+1;

```

```

        if(ehora == 100)

            ehora=0;

            perc

        end

    end

end

matriz_imagem = imread(Grid);

matriz_pontos = alvo; % Colocar todos os pontos que deseja pintar

espessura_linha = 1; % pinta so o pixel se quiser pintar ao redor do pixel tb é so
aumentar o numer 2, 3, 4 ...

r = 255;          % coordenada da parte vermelha

g = 0;

b = 0;

if(espessura_linha < 13)

    if ((espessura_linha > 0)&&(espessura_linha/(round(espessura_linha))))

        num_niveis = espessura_linha - 1;

        nlin_m_pontos = size(matriz_pontos,1); % Numero de linhas da matriz de pontos

        for k = 1 : nlin_m_pontos

            u = matriz_pontos(k,1);

            v = matriz_pontos(k,2);

            for cont_v = num_niveis*(-1):num_niveis

```

```

        for cont_u = num_niveis*(-1):num_niveis

            if (((u+cont_u)>=1) &&
((v+cont_v)>=1)&&((u+cont_u)<=(size(matriz_imagem,2))) &&
((v+cont_v)<=(size(matriz_imagem,1))))

                x=(u + cont_u);

                y=(v + cont_v);

                matriz_imagem(y,x,1) = r;

                matriz_imagem(y,x,2) = g;

                matriz_imagem(y,x,3) = b;

            end

            cont_u = cont_u + 1;

        end

        cont_v = cont_v + 1;

    end

    k=k+1;

end

end

end

matriz_imagem_s = matriz_imagem;
image(matriz_imagem_s);          % Usado so para ver a imag

```

Rotina Mostra Alvo

```
function mostra_alvo(C,nome_da_imagem)

Grid = strcat('D:/UFRJ/PF/final/',nome_da_imagem,'.bmp');

matriz_imagem = imread(Grid);

matriz_pontos = C; % Colocar todos os pontos que deseja pintar

espessura_linha = 3; % pinta so o pixel se quiser pintar ao redor do pixel tb é so aumentar
o numer 2, 3, 4 ...

r = 255; % coordenada da parte vermelha

g = 0;

b = 0;

if(espessura_linha < 13)

if ((espessura_linha > 0)&&(espessura_linha/(round(espessura_linha))))

num_niveis = espessura_linha - 1;

nlin_m_pontos = size(matriz_pontos,1); % Numero de linhas da matriz de pontos

for k = 1 : nlin_m_pontos

u = round (matriz_pontos(k,1));

v = round (matriz_pontos(k,2));

for cont_v = num_niveis*(-1):num_niveis

for cont_u = num_niveis*(-1):num_niveis

if (((u+cont_u)>=1) && ((v+cont_v)>=1)&&((u+cont_u)<=(size(matriz_imagem,2)))

&& ((v+cont_v)<=(size(matriz_imagem,1))))

x=(u + cont_u);

y=(v + cont_v);
```



```

if (x<1 || x>size(matriz_imagem,2)||y<1 ||y>size(matriz_imagem,1))

s = sprintf ('alvo invalido x=%d e y = %d',x,y);

disp(s);

else

matriz_imagem(y,x,1) = r;

matriz_imagem(y,x,2) = g;

matriz_imagem(y,x,3) = b;

end

end

cont_u = cont_u + 1;

end

cont_v = cont_v +1;

end

k=k+1;

end

else

end

else

end

matriz_imagem_s = matriz_imagem;

image(matriz_imagem_s); % Usado so para ver a imagem

```