



Universidade Federal de Mato Grosso do Sul  
Curso de Bacharelado em Sistemas de Informação

## **Segurança com Kinect**

**Andre Luiz Silva de Castro**

**Corumbá-MS, 2018**

Andre Luiz Silva de Castro

## **Segurança com Kinect**

**Trabalho de Conclusão de Curso** Monografia apresentada a Universidade Federal do Mato Grosso do Sul (UFMS) como um dos pré-requisitos para a obtenção do grau de Bacharel em Sistemas de Informação.

Universidade Federal de Mato Grosso do Sul- UFMS

Orientador: Murilo Oliveira Machado

Corumba-MS

2018

Andre Luiz Silva de Castro

## **Segurança com Kinect**

**Trabalho de Conclusão de Curso** Monografia apresentada a Universidade Federal do Mato Grosso do Sul (UFMS) como um dos pré-requisitos para a obtenção do grau de Bacharel em Sistemas de Informação.

Trabalho aprovado. Corumbá-MS, 01 de janeiro de 2014:

---

**Murilo Oliveira Machado**  
Orientador

---

**Caroline Paula Cellini**  
Convidado 1

---

**Bárbara Regina Barros**  
Convidado 2

Corumbá-MS  
2018

*Para minha família e amigos  
por sempre estarem comigo em todos os momentos.*

# AGRADECIMENTOS

Agradeço aos meus pais, Elizabeth da Silva e Reginaldo de Castro Filho, e toda a minha família pelo apoio nesses longos anos.

Em especial aos meus irmãos Alan Lucas Silva de Castro e Maria Eduarda Silva de Castro, por me darem apoio moral por tanto tempo e gastarem madrugadas sendo meus modelos de teste.

Agradeço ao meu orientador, Murilo Oliveira Machado, por todos os conselhos e aos professores do curso de Sistemas de Informação pelo conhecimento que me foi passado. Em especial o professor Rogério Gutts que começou me orientando neste trabalho e me deu a base para programar o kinect. Também as professoras da banca pela oportunidade de apresentar este trabalho.

E por fim meus amigos que sempre me impulsionaram. Principalmente Bruno Daniel que me emprestou seu kinect por quase 2 anos.

*'Bom mesmo é ir à luta com determinação,  
abraçar a vida com paixão,  
perder com classe  
e vencer com ousadia,  
porque o mundo pertence a quem se atreve  
e a vida é muito para ser insignificante.'*

*(Augusto Branco)*

**Abstract.** Esta monografia descreve tecnologias úteis em sistemas de segurança que buscam identificar posições de perigo e acionar um alarme com gestos pré classificados. Foram implementados modelos matemáticos para classificar as imagens obtidas com o sensor kinect, sendo que os resultados iniciais desta pesquisa mostraram-se promissores, e revelando que, a junção do Kinect com as Redes Neurais Artificiais podem ser eficientes em classificar diferentes posições em tempo real com um baixo custo computacional.

**Palavras-chaves:** Kinect. RNA. Sistemas de classificação de padrões.

**Abstract.** This monograph describes useful technologies in security systems that seek to identify hazardous positions and trigger an alarm with pre-classified gestures. Mathematical models were implemented to classify the images obtained with the kinect sensor, and the initial results of this research were promising, showing that the Kinect junction with Artificial Neural Networks can be efficient in classifying different positions in real time with a low computational cost.

**Keywords:** kinect. RNA. Pattern classification systems.

# LISTA DE ILUSTRAÇÕES

Figura 1 – Representação do Funcionamento de uma Rede Neural . . . . .	4
Figura 2 – Imagem que representa o treinamento do perceptron. Disponível em: <a href="https://pt.wikipedia.org/wiki/Perceptron_multicamadas/">https://pt.wikipedia.org/wiki/Perceptron_multicamadas/</a> . . . . .	5
Figura 3 – Formula de treinamento do perceptron . . . . .	6
Figura 4 – Imagem que representa a classificação do perceptron disponível em: <a href="https://pt.wikipedia.org/wiki/Perceptron_multicamadas">https://pt.wikipedia.org/wiki/Perceptron_multicamadas</a> . . . . .	6
Figura 5 – kinect disponível em: <a href="http://canaltech.com.br/o-que-e/kinect/Como-funciona-o-Kinect/">http://canaltech.com.br/o-que-e/kinect/Como-funciona-o-Kinect/</a> . . . . .	7
Figura 6 – Página Inicial do Sistema . . . . .	16

# LISTA DE TABELAS

Tabela 1 – Porcentagem de acertos obtidos para 5 arquiteturas de RNA com treinamento limitado em 8000 épocas . . . . .	17
------------------------------------------------------------------------------------------------------------------------	----

# LISTA DE ABREVIATURAS E SIGLAS

RNA	Rede Neural Artificial
MLP	Perceptron de Multicamadas
SVM	Maquinas de Vetores de Suporte

# SUMÁRIO

<b>Introdução</b>	1
<b>1 EMBASAMENTO TEÓRICO</b>	3
<b>1.1 RNAs</b>	3
<b>1.2 Notas históricas e primeiras RNAs</b>	4
<b>1.3 Características das RNAs</b>	5
<b>1.4 Modelos matemáticos para neurônios artificiais</b>	6
<b>1.5 Kinect</b>	7
<b>1.6 Hardware</b>	7
<b>1.7 Microfones</b>	8
<b>1.8 C Sharp</b>	8
<b>2 REVISÃO BIBLIOGRÁFICA</b>	9
<b>2.1 Mobile Robots Navigation in Indoor Environments Using Kinect Sensor</b>	9
<b>2.2 Towards Skeleton Biometric Identification Using the Microsoft Kinect Sensor</b>	10
<b>2.3 Human Gesture Recognition Using Kinect Camera</b>	11
<b>2.4 Real-Time Human Pose Recognition in Parts from Single Depth Images</b>	12
<b>3 METODOLOGIA</b>	14
<b>4 RESULTADOS E DISCUSSÃO</b>	16
<b>4.1 Sistema em C#</b>	16
<b>4.2 Métodos de Classificação</b>	16
<b>Conclusão e Trabalhos Futuros</b>	19

# INTRODUÇÃO

O desenvolvimento da tecnologia nas últimas décadas vem possibilitando grandes avanços tecnológicos, afetando diretamente todos os aspectos da vida humana [de Oliveira and Simões 2017]. Sua presença pode ser encontrada desde recreação ao trabalho, abrindo possibilidades para novas ideias e aplicações.

O sensor Kinect foi projetado para identificar os membros humanos e substituir controles dos video-games, possibilitando um novo modo do computador perceber o mundo ao seu redor [Miles 2012]. E junto com uma técnica de classificação de padrões, como redes neurais, pode ser utilizado para identificar movimentos ou posições que manifestem situações de insegurança, sendo uma boa opção para monitorar ambientes como visto em [Haykin 2007], [Silva et al. 2010].

A junção de câmera, sistema de microfone e sensor de profundidade em um único aparelho cria uma oportunidade para os desenvolvedores de softwares aumentarem as aplicações de interação entre o computador e o homem, de maneiras versáteis e com uma grande capacidade de processamento aliado a um baixo custo [Haykin 2007], [Patsadu et al. 2012].

Esta monografia apresenta fundamentos de um sistema de segurança com a finalidade de monitorar o ambiente fechado e identificar situações de perigo. Descreve a implementação de uma Rede Neural Artificial (RNA) e classificadores Suport Vector Machine(SVM) utilizando o sensor Kinect para capturar as imagens deste ambiente.

A motivação surgiu após a participação de um projeto de extensão que visava explorar e desenvolver aplicações com o Kinect. Depois de estudar RNAs e realizar uma pesquisa sobre trabalhos relacionados, foi constatado que era possível criar um sistema inteligente que pudesse identificar momentos de risco através de movimentos de pessoas, o que incentivou a criar um sistema capaz de monitorar um ambiente em tempo real e proporcionar ao usuário acionar um alarme, em caso de assalto, sem utilizar botões ou comandos de voz, apenas com uma posição pré-salva pelo sistema é possível acionar um alarme sonoro.

O objetivo da monografia consiste em desenvolver um sistema de segurança que possa atender as necessidades de segurança de lojas e empresas utilizando a tecnologia do sensor de movimentos do Kinect. E Utilizar modelos matemáticos eficientes em classificar imagens para monitorar o ambiente das lojas e identificar quando está ocorrendo um assalto

Foi realizado uma breve introdução do tema e objetivos da monografia, em seguida

será descrita uma contextualização histórica e os principais conceitos de uma RNA, do Kinect e uma revisão bibliográfica sobre alguns sistemas de segurança dentro dessa perspectiva. E por fim apresentada a metodologia utilizada para atingir o objetivo, discussão dos resultados e considerações finais.

# 1 EMBASAMENTO TEÓRICO

Neste capítulo será abordada a base teórica que levou a criação do sistema, sendo primeiro abordados os conceitos, história, funcionamento e benefícios de uma RNA. Em seguida o equipamento *kinect*, linguagem de programação C# e a plataforma *Visual Studio* utilizados para a criação do sistema.

## 1.1. RNAs

São modelos matemáticos auto-ajustáveis inspirados no processamento de informações visto nos neurônios, de forma geral foi projetado como sistema nervoso humano processa suas informações e realiza ações. Segundo [Haykin 2007, p.28] uma rede neural é um processador maciçamente distribuído e construído de unidades de processamento simples, que tem a propensão natural para armazenar conhecimento experimental e torná-lo disponível para uso [Silva et al. 2010].

Na sua forma mais geral, uma rede neural é uma máquina que é projetada para simular a maneira de como a mente humana processa e realiza suas ações. Para se obter um bom desempenho as redes neurais empregam uma interligação maciça de células computacionais simples denominadas “neurônios” ou “unidades de processamento”.

O sistema nervoso humano pode ser visto em três estágios, a figura 1 representa o processamento de informações pelo cérebro humano. Os receptores convertem estímulos do corpo ou ambiente externo em impulso elétricos que transmitem as informações para a rede neural (cérebro), e em seguida os atuadores convertem impulsos elétricos gerados pela rede neural em respostas consideradas saídas.

Um neurônio em “desenvolvimento” é sinônimo de um cérebro plástico: a plasticidade permite que o cérebro se adapte ao seu meio ambiente. Assim como a plasticidade parece ser essencial para o funcionamento do cérebro humano, também é em relação às RNAs.

Normalmente, os neurônios são cinco a seis ordens de grandeza mais lentas que as portas lógicas em silício; os eventos em um circuito de silício ocorrem em nanosegundos enquanto os do cérebro humano ocorrem em milisegundos. Entretanto, o cérebro compensa a falta de velocidade de processamento com uma quantidade bem elevada de conexões sinápticas.

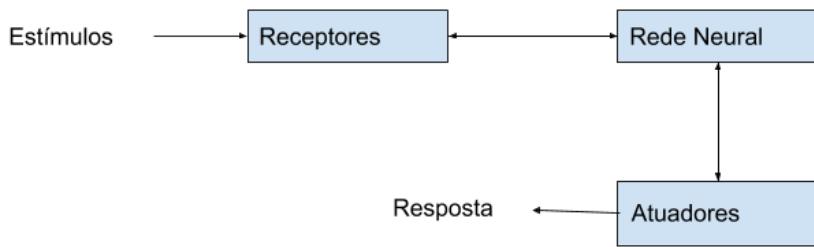


Figura 1: Representação do Funcionamento de uma Rede Neural

## 1.2. Notas históricas e primeiras RNAs

As primeiras informações sobre a criação de RNAs foram apresentadas em 1943 no artigo “Neurônios Formais” do neuroanatomista e psiquiatra Warren McCulloch e do matemático Walter Pitt em 1943. Posteriormente Wiener contribui com as RNAs publicando em 1948 o livro Cybernetics, que descreve que o cérebro está em constante modificação, pois o organismo sempre procura adaptar (aprender) a novas tarefas a ele imposta.

Em 1958 Rosenblatt escreveu o teorema de convergência do perceptron, uma grande evolução para o problema de reconhecimento de padrões, onde ele descrevia um modelo cognitivo que consistia de unidades sensoriais conectadas a uma única camada de neurônios de Warren McCulloch e Pitts, capaz de aprender tudo o que pudesse representar, demonstrando que se fossem adicionadas sinapses ajustáveis, as redes neurais de Warren McCulloch e Pitts poderiam ser treinadas para classificar padrões em classes linearmente separáveis.

O Perceptron é um modelo matemático que representa um neurônio com a função de ativação não linear, que no seu treinamento realiza ajustes em variáveis conhecidas como pesos sinápticos, aprendendo a diferenciar entradas em dois grupos diferentes, sendo capaz de resolver problemas linearmente separáveis, podendo ser considerado um algoritmo de classificação binário.[[Haykin 2007](#)].

Em 1960 Widrow e Hoff apresentaram o algoritmo de treinamento utilizando o erro mínimo quadrado médio, e posteriormente incrementado em um modelo com a função de propagação linear (Adaline), e culminando a regra delta de aprendizagem.

Na década de 70 não foram escritos muitos artigos ou livros sobre RNA por este modelo matemático estar desacreditado no meio acadêmico, onde surgiram afirmações exageradas por parte dos estudiosos, que publicaram artigos e dissertações afirmando previsões que para a época eram irreais, como o aparecimento de máquinas tão potentes quanto a mente humana em pouco tempo em conjunto com a falta de tecnologia da época.

O Adaline ( Neurônio Adaptativo Linear) foi criado por Widrow e Hoff após

o surgimento do perceptron em 1959, e fornece saídas lineares usadas para classificar padrões, ajusta os pesos sinápticos utilizando a regra Delta para minimizar a média dos erros após cada padrão apresentado. O Perceptron de Multicamadas (MLP) é uma evolução do perceptron comum possuindo mais camadas de neurônios. O aprendizado da MLP é realizado pelo algoritmo de treinamento backpropagation. [Haykin 2007]

### 1.3. Características das RNAs

Uma rede neural extrai seu poder computacional através de sua estrutura maciça-mente paralela e distribuída, tem a capacidade de aprender e generalizar o conhecimento. A generalização se refere ao fato de a RNA produzir saídas adequadas para entradas que não estavam presentes durante seu treinamento, portanto permitem resolverem problemas que são atualmente intratáveis[Haykin 2007]. A figura 2 mostra a superfície de decisão para dois problemas, um linearmente separável e outro não.

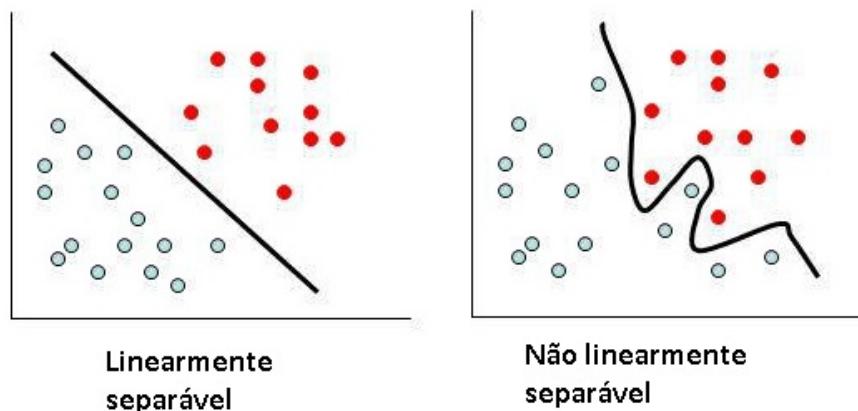


Figura 2: Imagem que representa o treinamento do perceptron. Disponível em: [https://pt.wikipedia.org/wiki/Perceptron\\_multicamadas/](https://pt.wikipedia.org/wiki/Perceptron_multicamadas/)

Como dito no começo do capítulo, RNAs e modelos de classificação são baseadas no funcionamento do cérebro humano, e como tal é preciso de conhecimento prévio para executar tarefas com exatidão. No treinamento de uma RNA, é necessário ter uma saída desejada para possíveis entradas, como por exemplo no desenvolvimento deste trabalho, que foram tiradas fotos de pessoas com as mãos tanto para cima quanto para baixo. Os dados destas fotos foram utilizadas como base para a aprendizagem da rede neural, assim, com um conhecimento prévio de que em uma foto a pessoa está com as mãos levantadas e em outra está com elas abaixadas, pode diferenciar outras imagens que não foram utilizadas no seu treinamento e assim diferencia-las.

Uma RNA é eficiente em classificar padrões, tornando-as importantes em casos ambíguos que se referem a mesma situação. Cada neurônio da rede é afetado pelas ações

de todos os outros neurônios na rede, com isso as informações contextuais são tratadas naturalmente pela rede. Esta padronização torna possível que sejam compartilhadas teorias de redes neurais para vários problemas diferentes utilizando um mesmo método.

Uma rede neural implementada em hardware tem o potencial de ser tolerante a falhas, no sentido de que seu desempenho não diminui em caso de situações adversas, como o mau funcionamento de parte do hardware. Por ser basicamente paralelas, as redes neurais são potencialmente melhores em certas tarefas, também a se mostra adequada para ser implementada em uma tecnologia de integração muito mais ampla.

#### 1.4. Modelos matemáticos para neurônios artificiais

Um neurônio é uma unidade de processamento de informação para a operação de uma rede neural. A figura 3 mostra um modelo de neurônio, que forma a base para uma rede neural. Nela é possível ver três elementos básicos de um modelo neural :

$$E \begin{cases} x_1 \rightarrow w_1 \\ x_2 \rightarrow w_2 \\ ..... \quad ..... \\ x_n \rightarrow w_n \end{cases} \rightarrow v_k = \sum_{i=0}^n \omega_i x_i \rightarrow f(v_k) \rightarrow sada$$

Figura 3: Formula de treinamento do perceptron

O conjunto de elos de conexão ou sinapses possuem um peso próprio  $w_i$ . Se pegarmos uma variável qualquer do conjunto, por exemplo a sinapse  $x_i$ , ela será multiplicado pelo peso sináptico de respectivo neurônio  $w_i$ , sendo o somatório de todas sinapses é utilizado como entrada da função de ativação fornecendo a saída da rede.

É possível melhorar o modelo matemático da RNA descrito na figura x, acrescentando um termo independente obtendo a relação  $v_k = \sum_{i=0}^n \omega_i x_i + b$ , onde b é chamado de bias. E Dependendo se a bias é positivo ou negativo a relação entre o campo local induzido de vK do neurônio K e a saída U é modificada como pode ser visto na figura 4:

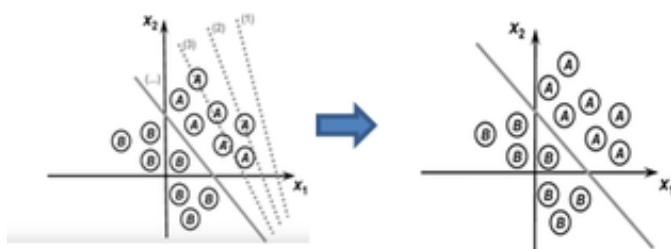


Figura 4: Imagem que representa a classificação do perceptron disponível em: [https://pt.wikipedia.org/wiki/Perceptron\\_multicamadas](https://pt.wikipedia.org/wiki/Perceptron_multicamadas)

## 1.5. Kinect

É um sensor de movimento desenvolvido pela Microsoft e lançado no ano de 2009, conhecido como projeto natal. Segundo [Miles 2012], até recentemente os computadores possuíam uma visão muito restrita do mundo ao seu redor, e usuários tinham muita limitação ao se comunicar com computadores. Ao longo dos anos os computadores adquiriram câmeras e entradas de áudio, mas eles foram utilizados na maioria das vezes para entradas que não eram reconhecidas; computadores podem armazenar e reproduzir vários conteúdos, mas vem sendo bem difícil de fazer os computadores entenderem essas entradas. [Patsadu et al. 2012] fizeram uma pesquisa avaliando diversas formas para rastrear movimentos humanos. Em sua dissertação eles afirmam que utilizam o kinect como ferramenta para processar a captura de movimentos pela seu baixo custo e sua versatilidade.

Para [Miles 2012] o kinect mudou a maneira de como os computadores interagem com o mundo. A barra do sensor kinect contém duas câmeras, uma fonte especial de luz infravermelho e quatro microfones. Ele também contém uma pilha de sinais de processamento de hardware que o faz ser capaz de processar toda a informação que as câmeras, luzes infravermelhas e microfones podem gerar. Combinando a saída desses sensores , o programa pode localizar e reconhecer objetos à sua frente, determinar a direção dos sinais de som, e os isolador de ruídos de fundo.



Figura 5: kinect disponível em:<http://canaltech.com.br/o-que-e/kinect/Como-funciona-o-Kinect/>

## 1.6. Hardware

O sensor de profundidade é uma ferramenta importante no desenvolvimento deste trabalho, uma vez que, com ele será feita a leitura do meio ambiente para o funcionamento da aplicação. Segundo [Miles 2012] o kinect tem a habilidade única de ver em 3d. Diferente dos outros sistemas de visão de computador, o sensor kinect pode criar um mapa de profundidade da área à sua frente. Este mapa é produzido inteiramente com a barra do sensor e é transmitido para o host via cabo USB do mesmo jeito do que uma câmera de imagem transmitirá, exceto que ao invés de informação de cor para cada pixel em uma imagem, o sensor transmite valores de distância.

## 1.7. Microfones

O kinect também possui um sistema de captura de voz que é utilizado para se dar comandos em jogos ou para navegar no console sem precisar de controle. Segundo [Miles 2012] o sensor kinect também possui quatro microfones ao longo da parte inferior da barra, dois na parte esquerda e mais dois na parte direita. O kinect usa esses microfones para ajudar de onde na sala uma determinada voz está vindo. Isto funciona porque o som leva tempo para viajar pelo ar. O som viaja muito mais lentamente do que a luz, é por isso você ouve o som do relâmpago muito depois da queda.

A câmera e o sensor de profundidade são parte principal do projeto, pois com eles será feita a leitura dos movimentos e a captura dos movimentos para o funcionamento do sistema mesmo durante o período noturno.

## 1.8. C Sharp

Segundo [DURAES 2008] o C# é uma linguagem de programação criada para o desenvolvimento de uma variedade de aplicações que executam sobre o .NET Framework. C# é uma linguagem simples, mas poderosa, e orientada a objetos. O C# permite o desenvolvimento rápido de aplicações, mantendo a simplicidade das linguagens C. Visual C# é uma implementação da linguagem C# pela Microsoft.

Esta é uma linguagem de programação que é utilizada para executar comandos no kinect. Com ela junto ao kinect é possível fazer a varredura do local e interpretar os gestos e movimentos de futuras ameaças. Para desenvolver aplicações em C# o livro de [Miles 2012] indica o visual studio como ferramenta de desenvolvimento, pela comodidade e eficiência que oferece ao seu usuário.

O visual studio é a IDE (Ambiente de desenvolvimento integrado) que foi utilizada para o desenvolvimento do sistema. Segundo [DURAES 2008] o ambiente de desenvolvimento integrado do Visual Studio é um painel de inicialização criativo que você pode usar para editar, depurar e compilar o código e, em seguida, publicar um aplicativo.

Além de ser um editor e possuir um depurador, ele ainda possui um compilador, junto com ferramentas de preenchimento de código e ferramentas de design gráfico, para facilitar o desenvolvimento de sistemas.

## 2 REVISÃO BIBLIOGRÁFICA

Nesta capítulo serão discutidos os trabalhos que foram utilizados como base para o desenvolvimento desta monografia, e tem como foco pesquisas que utilizam o kinect em conjunto com métodos de classificação de padrões para uso em aplicações, como navegação robótica, sistemas biométricos, reconhecimento de posições. Serão abordados os sistemas que utilizaram como base as redes neurais combinadas com o sensor kinect.

### **2.1. Mobile Robots Navigation in Indoor Environments Using Kinect Sensor**

Na área de segurança há sistemas que fazem vigilância residenciais em locais fechados, conhecido também como sistema de segurança indoor, que utilizam o kinect como principal ferramenta de vigilância de seus sistemas. O trabalho de [Correa et al. 2012] utiliza mini robôs equipados com os sensores de distância do kinect.

No seu trabalho Mobile Robots Navigation in Indoor Environments Using Kinect Sensor, escrito em 2012, descreve a criação de uma aplicação de segurança para ambientes fechados utilizando um robô móvel que evita obstáculos utilizando o sensor de profundidade do kinect acoplado a ele. O sistema é dividido em duas partes distintas. A primeira é a parte física que conta com o robô e o kinect e a segunda é a criação de uma rede neural Artificial para analisar o ambiente e “decidir” quando o robô deve se desviar de um obstáculo.

Foi destacada a importância de robôs autônomos para a sociedade moderna como uma forma de segurança. Na construção desses robôs é preciso se ter em mente a plataforma e os sensores que compõem o hardware. Os sensores são utilizados para coletar informações sobre o ambiente, mas para o pleno funcionamento é preciso resolver três problemas principais: mapear o ambiente, localizar e movimentar o robô. Depois é apresentado uma comparação entre sensores de segurança que chega à conclusão de que o kinect é uma ferramenta mais eficaz por ser mais barato e potente do que outros sensores como por exemplo o hokuyo laser. Sendo descritas algumas razões para o avanço no desenvolvimento de robôs de vigilância, como a segurança dos guardas noturnos e a realização de tarefas que são monótonas para seres humanos.

No desenvolvimento foi apontado que o sistema é composto de duas partes e que cada uma foi testada como dois sistemas separados para validar seu funcionamento. O subsistema de percepção [1] recebe informações da câmera 3d e a câmera de profundidade do kinect e cria um mapa de navegação do local. Esse mapa é utilizado para que o robô ande pelo ambiente e evite obstáculos. O segundo subsistema [2] utiliza processamento de

imagens e técnicas de inteligência artificial para sobre os dados retirados do kinect para classificar os dados do ambiente para localizar o robô dentro dele.

## 2.2. Towards Skeleton Biometric Identification Using the Microsoft Kinect Sensor

No seu artigo [[Araujo et al. 2013](#)] propos a utilização do sensor kinect para extrair pontos do esqueleto para serem utilizados em identificação biométrica. Os pontos são identificados através do comprimento dos membros do corpo humano que são extraídos pelo sensor e depois treinados em um modelo de RNA para classificar cada pessoa cadastrada através da estrutura de seu corpo.

Foi considerado um caso em que era necessário discriminar apenas uma pessoa individualmente para o funcionamento do algoritmo de vizinho mais próximo, mostrando que essa abordagem é capaz de alcançar uma alta precisão quando se utiliza um grupo pequeno de testes. Entretanto esta abordagem precisa de uma grande quantidade de amostras de treinamento e é discutido o impacto que isso causa em alguns cenários.

Os autores afirmam que a identificação individual automática de pessoas é um Pré-Requisito para várias aplicações, principalmente para área de segurança onde o acesso a recursos são permitidos apenas a pessoas pré-cadastradas ou em sistemas que são customizadas para cada usuário. Com isso eles destacam que a identificação biométrica é um modo conveniente para cumprir esta tarefa, já que com ela é descartado o uso de senhas ou tokens de segurança (chaves ou cartões) com adição de que o usuário precisa estar fisicamente no local de identificação.

Sistemas biométricos são mais conhecidos como sistemas de reconhecimento de padrões que guardam informações de atributos físicos das pessoas de interesse e posteriormente identifica-lo por seus dados. Podem ser divididos em duas maneiras, uma que necessita que a pessoa interaja com uma interface de algum modo, como no caso de reconhecimento a partir da digital dos dedos, o outro é conhecido como sistemas de reconhecimento passivo, não necessitam contato direto com nenhuma interface, como o reconhecimento de voz e de face, como a tecnologia utilizada atualmente para destravar celulares.

Uma abordagem dos sistemas de reconhecimento passivo é utilizar várias partes do corpo humano, como peso ou a largura dos braços e pernas, que podem ser utilizados para reconhecer pessoas através de fotos e vídeos. No fim da introdução é informado que eles pretendem criar um sistema de reconhecimento biométrico passivo que identifique uma pessoa através da largura de seus membros, que foram retirados a partir de frames de video do kinect.

Os trabalhos relacionados descrevem uma breve história sobre o kinect e que desde sua criação foi utilizado em várias aplicações de reconhecimento fora de seu objetivo primário, que era ser um substituto de controles de videogame. Depois é descrito que a identificação biométrica de partes do corpo era comum em investigações criminais mas foi substituída por reconhecimento das digitais dos dedos, por demandar muito tempo e ser propenso a mais erros humanos. Mas com a popularização de câmeras e máquinas fotográficas ouve um aumento de interesse nesta área novamente.

O objetivo principal do artigo foi provar a viabilidade da utilização dos dados do kinect para a criação de um sistema biométrico passivo e utilizar vários modelos de aprendizado de máquinas para classificar os dados. Com a utilização de 8 voluntários (2 mulheres e 6 homens) foi capturado dados dos braços e antebraços em conjunto com as pernas e coxas em três coordenadas espaciais(x,y,z) para calcular a largura dos membros e os pontos da espinha para calcular a altura do voluntário. Depois de obter os dados foi realizado a identificação de cada um, para isso foram utilizadas técnicas de aprendizagem de máquinas como mlp, arvore de decisão, K-ésimo Vizinho mais Próximo e florestas aleatórias, que após os treinos revelaram 83%, 92%, 99.75 e 97.5% de precisão respectivamente.

### **2.3. Human Gesture Recognition Using Kinect Camera**

No artigo [Patsadu et al. 2012] descreve o reconhecimento de gestos humanos usando métodos de classificação de mineração de dados em streaming de vídeo. Utiliza o sensor kinect, para capturar posições, em conjunto com Métodos de classificação, o trabalho pretende identificar através de padrões as posições de em pé e sentado em seus voluntários. Os métodos de classificação escolhidos foram Redes neurais com Retro-propagação, SVM (Máquinas de vetor de suporte), árvores de decisão e naive bayes.

Foi discutido os conceitos básicos de gestos do corpo humano e sua importância nos trabalhos de segurança, detecção de movimentos e esportes. Depois realizado um estudo de vários métodos de classificação de dados e uma comparação entre eles. Logo após é feita uma descrição dos dados e como eles são retirados a partir de vetores que armazenam as posições do corpo da versão básica do kit de desenvolvimento de software do kinect.

Os autores destacam a evolução da tecnologia de vigilância com câmeras até chegar em câmeras que criam esqueletos em 3d. Com isso o ele passa a descrever o trabalhos de classificação de movimentos humanos em 3d que o levaram a escolher os métodos de classificação de padrões em seu trabalho e a criação do conjunto de amostras para o treinamento.

Foi descrito que o projeto se baseia em armazenar partes de vídeos em quadros estáticos, que são representados por um nó de vetor com vinte posições (x,y,z), para serem

utilizados posteriormente para treinar métodos de classificação de padrões e averiguar qual deles é o mais eficiente. Após o treinamento foi concluído que o svm é o que possui a maior porcentagem de acertos com 99,75%.

## 2.4. Real-Time Human Pose Recognition in Parts from Single Depth Images

O artigo de [Shotton et al. 2011] propos um novo método mais rápido e eficiente de prever juntas do corpo através de imagem de profundidade. Designando partes do corpo humano é criado um algoritmo que mapeia a dificuldade de generalizar poses e o transforma em um simples problema de classificação de pixel por pixel utilizando métodos de classificação de objetos. A grande quantidade de amostras e variedade de possíveis treinamentos possibilita classificar partes do corpo indiferente para poses, roupas e tamanho do corpo. Os resultados apresentaram uma grande precisão em testes sintéticos e reais e investigam os efeitos de vários parâmetros de treinamento.

Os autores afirmam que aplicações robustas que utilizam gestos humanos como controle são encontradas em video games, segurança, interação humano computador,etc. Esta tarefa foi suavizada com a introdução de câmeras de profundidade em tempo real, no entanto todos os sistemas que as utilizavam possuíam alguma limitação. Até a criação do kinect nenhum hardware de interação desenvolvido para o publico possuía uma gama completa de taxas interativas enquanto lida com os tamanhos e formas do corpo humano. Logo após essa introdução é descrito que o objetivo do trabalho é reconhecer posições de indivíduos a partir de imagens de profundidade que são tiradas a partir do kinect.

Foram apresentadas resumidamente pesquisas que descrevem a evolução de câmeras de profundidade e reconhecimento de pontos do corpo com 3d utilizando o skeleton e outros projetos que buscam fazer reconhecimento de padrões utilizando técnicas similares á proposta do autor.

É descrito que varias pesquisas na área de estimativa de poses foram feitas para sobrepor a dificuldade de dados que podem ser treinados, uma vez que com as câmeras comuns, as imagens são reduzidas a silhuetas em 2D, e o treinamento é dificultado pela cores das roupas,cabelo e cor da pele. Por outro lado as câmeras de profundidade reduzem estas dificuldades, pois conseguem diferenciar as variações das roupas e do corpo.

O autor afirma que a imagem em profundidade alcançou um novo patamar com a chegada do kiekt, que disponibilizou essa tecnologia para o publico geral a um preço baixo. As câmeras de profundidade tem uma grande vantagem em relação aos sensores comuns, podendo trabalhar com pouca luz, podem ser calibradas, elas desconsideram variáveis de cor e textura e resolvem o problema de poses ambíguas.

Foram apresentados pontos chave do corpo e depois o treinamento de um método

de classificação de padrões conhecido como florestas de decisão, que são treinadas com mais de 2000 pixels, para classificar várias posições diferentes do corpo humano. Os resultados apresentados mostram que os testes realizados pelo autor possuem uma crescente de eficiência com o aumento de amostras para classificação na classificação dos pontos do corpo humano.

### 3 METODOLOGIA

No primeiro estágio do desenvolvimento do trabalho foi feita uma pesquisa sobre trabalhos relacionados ao objetivo proposto, para descobrir a viabilidade da ideia. Após o término da coleta de informações, foi constatado que para construir o sistema seria preciso o entendimento de redes neurais e conhecimento da linguagem de programação c#, para poder programar o kinect.

O primeiro passo seguido foi o estudo da linguagem c# e o kinect. Para isso foi utilizado o livro *Start Here! Learn the Kinect API* de [Miles 2012]. No livro é detalhado os requisitos para criar aplicações com o kinect e exemplos práticos para se basear. O livro sugere que seja utilizado o visual studio, que é uma ferramenta que auxilia o desenvolvimento de códigos em c#. As primeiras versões do sistema em c# demoraram para serem feitas por causa da grande demandada de tempo requeridas e pelas seguidas falhas devido a inexperiência com o material. Foi preciso ler trabalhos acadêmicos que falavam sobre o assunto em conjunto com videoaulas e auxilio de fóruns na internet para ter uma base para a criação do trabalho.

Foi implementado posteriormente um algoritmo, que utiliza o kinect, para coletar amostras em tempo real. O algoritmo coleta amostras das coordenadas cartesianas referente as posições das mãos, pulsos, braços, cotovelos, ombros, quadril e cabeça, e cada ponto é representado por 3 números ( largura, profundidade e altura). No começo dos testes o algoritmo armazenava apenas as posições da cabeça e das duas mãos, mas posteriormente, com o aumento dos sucessos, essa quantidade foi aumentada para 13 posições.

A próxima etapa era treinar os dados coletados com métodos de classificação. Como visto em Revisão bibliográfica, existem diferentes métodos para classificação de padrões, e para diferenciar se as mãos estão para cima ou para baixo, foi escolhido o Perceptron e Perceptron de Multicamadas com a função de ativação Sigmoide, e um classificador *Support Vectors Machine*.

A primeira tentativa de treinar as redes foi feita utilizando a linguagem de programação Python, porem com a demanda de tempo para estuda-la logo foi descartada, dando lugar para um algoritmo em c utilizando a biblioteca encog. Esta biblioteca se mostrou bem útil para os objetivos deste trabalho possibilitando a criação e treinamento das redes de uma maneira rápida e fácil. Com esta biblioteca foram criadas as redes neurais e o classificador svm mostrou os melhores resultados.

Para finalizar foram utilizadas as redes criadas treinadas com um sistema que

monitoram o ambiente em tempo real. Para isso foi modificado o sistema para acusar quando as mãos estavam para cima e para baixo.

# 4 RESULTADOS E DISCUSSÃO

Neste capítulo serão abordados os resultados obtidos com o trabalho. Como o sistema proposto é dividido em duas partes distintas, eles foram avaliados separadamente. Primeiro será apresentado o sistema desenvolvido em c#, e posteriormente os resultados de generalização dos métodos de classificação utilizados para discriminar as posições de perigo.

## 4.1. Sistema em C#

Como descrito na metodologia foi primeiramente construído uma aplicação que monitora o ambiente, em tempo real, para encontrar um usuário e depois salvar as suas posições para o treinamento dos métodos de classificação.

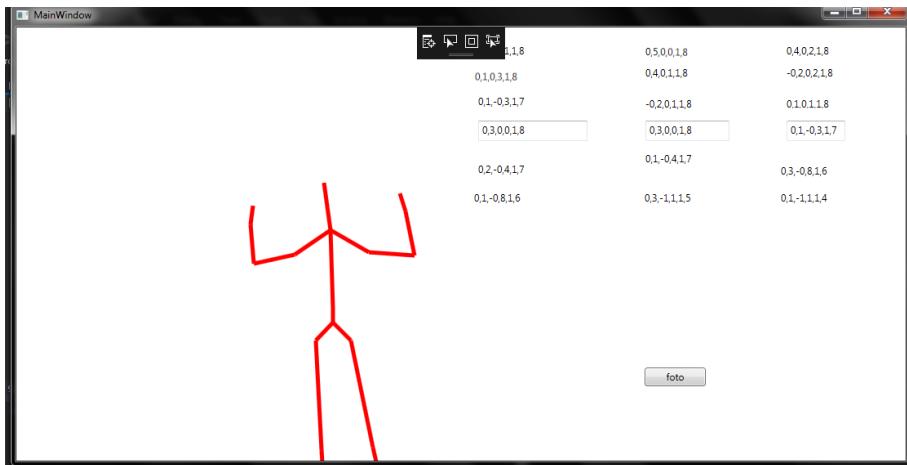


Figura 6: Página Inicial do Sistema

Como mostrado na figura 7, quando o kinect encontra uma pessoa ele começa a construir uma representação de cada parte do corpo dela. Cada parte do corpo está possuí coordenadas cartesianas que representam o seu posicionamento em relação ao kinect. Esses pontos são representados pelos números na parte superior direta da imagem acima.

Quando clicado no botão foto, os pontos são salvos e armazenados em um vetor que é gravado em um arquivo .txt, tornando assim possível salvar múltiplas posições em um pequeno espaço de tempo, diminuindo o tempo para conseguir amostras.

## 4.2. Métodos de Classificação

Os testes iniciais desta pesquisa investigaram a eficiência para diferentes tamanhos de amostras no treinamento e quantidade de neurônios em suas camadas, por fim mensurou

sua generalização para uma quantidade de amostras do mesmo tamanho e não vista no treinamento. No treinamento foi utilizado um conjunto de 10, 20, 40, 50 amostras com 13 pontos para definir os membros do corpo, que foram separadas em duas classes diferentes definidas a partir da altura das mãos. Dessa forma, as Redes Neurais Artificiais construídas são compostas de 39 entradas e duas saídas.

A tabela 1 mostra a generalização dos testes realizados para cada arquitetura de uma Rede Neural Artificial, seguindo a respectiva ordem, Perceptron (1), Perceptron de Camada Única com 5 neurônios (2), Perceptron de Camada Única com 10 neurônios(3), Múltiplo Perceptron de duas camadas com 5 neurônios(4), Múltiplo Perceptron de duas camadas com 10 neurônios (5).

Tabela 1: Porcentagem de acertos obtidos para 5 arquiteturas de RNA com treinamento limitado em 8000 épocas

arquitetura	10 amostras	20 amostras	40 amostras	50 amostras
1	20%	50%	50%	60%
2	60%	90%	90%	80%
3	40%	30%	80%	80%
4	60%	70%	80%	80%
5	30%	30%	30%	90%

Pode-se observar na tabela 1 que a generalização é afetada pela quantidade de amostras no treinamento, e o Perceptron não obteve mais que 60 por cento de acerto, isso pode ser justificado pela função de ativação não linear e baixa quantidade de amostras no treinamento para um conjunto grande de entradas nem sempre linearmente separáveis.

Não esperava-se melhores resultados obtidos com o Perceptron em relação arquitetura (2) como visto para 20 amostras de treinamento, bem como, para a arquitetura (5) com treinamento de 20 e 40 amostras, esse resultado foi atribuído a qualidade das amostras utilizadas nesse treinamento.

Observa-se que a arquitetura (5) possui um resultado ruim com menos de 40 amostras, contudo teve uma diferença significativa com 50 amostras no treinamento, 90 % de precisão, obtendo um resultado maior do que a arquitetura (2), que era a arquitetura mais precisa desta preciso até então.Pode-se afirmar que entre as diferentes arquitetura investigadas, o melhor resultado foi obtido com arquitetura (2), perdendo unicamente para a arquitetura (5) com 50 amostras de treinamento, que atingiu uma porcentagem de 90% de acertos, porem a arquitetura (2) possui o melhor resultado com menos de 50 amostras.

O classificador SVM foi treinado com o mesmo objetivo que as RNA (classificar mãos para cima e para baixo respectivamente), recorrendo a 13 pontos, 39 entradas, e mesma quantidade de amostras que as Redes Neurais Artificiais, e o melhor resultado

generalizou 80% para amostras não vista no treinamento.

Esses resultados são preliminares e serviram para investigar a generalização em função da quantidade de amostras utilizadas no treinamento de diferentes arquiteturas de RNA. E vale ressaltar que o kinect é uma tecnologia descontinuada e foi utilizado por conveniência, visto que a faculdade de Sistemas de Informação disponibilizou um e existe uma vasta literatura a respeito com bons resultados.

# CONCLUSÕES E TRABALHOS FUTUROS

## Considerações finais

O sistema com o kinect se provou capaz de identificar as posições de pessoas utilizando uma rede neural. Porém sua precisão está vinculada com a da rede neural, ou seja, para melhorar sua eficiência será preciso uma rede neural mais robusta, em questão de amostras e precisão.

Pode-se concluir que a quantidade de pontos e amostras são fundamentais para o desempenho da RNA e que a função de propagação Sigmoide teve o melhor resultado em relação a generalização para as amostras testes e tempo de convergência do treinamento das RNAs, por outro lado o classificador SVM teve o melhor desempenho em tempo de convergência em sua classificação.

Pretende-se com trabalhos futuros aperfeiçoar a Rede Neural Artificial, realizando um tratamento mais rigoroso com as amostras, acrescentar mais duas posições para serem classificadas, e utilizar no treinamento a partir de 300 amostras. Posteriormente customizar o sistema c# para que o usuário final possa ele mesmo cadastrar as posições que melhor lhe agrade e treine o sistema sem nenhum auxílio de terceiros.

## Referências

- Araujo, R. M., Graña, G., and Andersson, V. (2013). Towards skeleton biometric identification using the microsoft kinect sensor. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, pages 21–26. ACM.
- Correa, D. S. O., Sciotti, D. F., Prado, M. G., Sales, D. O., Wolf, D. F., and Osorio, F. S. (2012). Mobile robots navigation in indoor environments using kinect sensor. In *2012 Second Brazilian Conference on Critical Embedded Systems*, pages 36–41. IEEE.
- de Oliveira, F. T. and Simões, W. L. (2017). A indústria 4.0 e a produção no contexto dos estudantes da engenharia. In *Simpósio de engenharia de produção*.
- DURAES, R. (2008). *Desenvolvendo para web usando o Visual Studio 2008*. Brasport.
- Haykin, S. (2007). *Redes neurais: princípios e prática*. Bookman Editora.
- Miles, R. (2012). *Start Here! Learn the Kinect API*. Microsoft.
- Patsadu, O., Nukoolkit, C., and Watanapa, B. (2012). Human gesture recognition using kinect camera. In *Computer Science and Software Engineering (JCSSE), 2012 International Joint Conference on*, pages 28–32. IEEE.
- Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., and Blake, A. (2011). Real-time human pose recognition in parts from single depth images. pages 1297–1304.
- Silva, I. d., Spatti, D. H., and Flauzino, R. A. (2010). Redes neurais artificiais para engenharia e ciências aplicadas. *São Paulo: Artliber*, 23(5):33–111.