

Unioeste - Universidade Estadual do Oeste do Paraná
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
Colegiado de Ciência da Computação
Curso de Bacharelado em Ciência da Computação

**Prototipação de um sistema IoT de Reconhecimento de Ervas Daninhas e
Irrigação Artificial**

Fernando Stopasola Viera

CASCADEL
2019

Fernando Stopasola Vieira

Prototipação de um sistema IoT de Reconhecimento de Ervas Daninhas e Irrigação Artificial

Monografia apresentada como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação, do Centro de Ciências Exatas e Tecnológicas da Universidade Estadual do Oeste do Paraná - Campus de Cascavel.

Orientador: Prof. Dr. Marcio Seiji Oyamada

CASCADEL
2019

Fernando Stopasola Vieira

Prototipação de um sistema IoT de Reconhecimento de Ervas Daninhas e Irrigação Artificial

Monografia apresentada como requisito parcial para obtenção do Título de Bacharel em Ciência da Computação, pela Universidade Estadual do Oeste do Paraná, Campus de Cascavel, aprovada pela Comissão formada pelos professores:

Prof. Dr. Marcio Seiji Oyamada (Orientador)
Colegiado de Ciência da Computação,
UNIOESTE

Prof. Dr. Adair Santa Catarina
Colegiado de Matemática, UNIOESTE

Prof. M.Sc. Edmar Bellorini
Colegiado de Ciência da Computação,
UNIOESTE

Cascavel, 20 de dezembro de 2019

AGRADECIMENTOS

Agradeço minha família, pelo suporte necessário durante todas as etapas da graduação. Em especial a minha mãe Lucinara, que nunca mediu esforços para me ajudar durante todos os momentos que precisei. A minha banca por disponibilizar seu tempo para avaliar este trabalho. A meus amigos que tornaram o ambiente universitário mais descontraído e tranquilo (hehe) durante a jornada que passamos. A meu orientador Márcio que sempre me ajudou e orientou de maneira excepcional durante este projeto. Por fim, agradeço a Renata Alcântara que sempre me ajudou e motivou a finalizar o curso.

Lista de Figuras

2.1	Modelagem do robô.	5
2.2	Aplicação web para um módulo FarmBot.	6
2.3	Peça responsável pela injeção de sementes.	7
2.4	Diagrama de Arquitetura.	8
2.5	Design Sistema.	9
2.6	Unidade WSU.	10
2.7	Unidade central WIU.	11
2.8	Design Sistema.	12
2.9	Fluxograma do sistema.	14
2.10	Sensor vazão de água.	15
2.11	Webpage do sistema.	16
2.12	Diagrama de blocos do sistema.	18
2.13	Pseudocódigo segmentação baseado em textura.	20
2.14	Pseudocódigo Algoritmo reconhecimento folhas.	20
2.15	Processo de classificação de ervas daninhas.. . . .	21
2.16	Interface de detecção semiautônoma.	23
2.17	Imagem pré-processada.	25
2.18	Formulas usadas no Pré-Processamento	25
2.19	Representação Gráfica do Hiperplano.	26
2.20	Arquitetura da ANN.	27
2.21	Arquitetura da CNN.	28
2.22	Gráfico resultados CNN.	29
3.1	Arquitetura do sistema.	32

3.2	Estrutura de armazenamento do sistema.	33
3.3	Circuito para o controle motor de passo.	35
3.4	Imagem do sistema.	36
3.5	Circuito do sistema.	36
3.6	Sensor de Umidade.	37
3.7	Sistema de Visão computacional (SVC) e suas principais etapas.	38
3.8	Resultado Pré-Processamento.	39
3.9	Resultado Segmentação.	40
3.10	Ilustração <i>Local Binary Patterns</i>	41
3.11	Tuíte status do sistema.	43
3.12	Sensores danificados.	44
3.13	Falha segmentação.	47
3.14	Combinações entre imagens analisadas.	48

Lista de Abreviaturas e Glossário

TM4C123GH6PM	Microcontrolador
Hz	Hertz
dB	Decibel
WSU	Wireless Sensor Unit (unidades com sensor wireless)
WIU	Wireless Information Unit (unidade de informação wireless)
WSN	Rede de sensores sem fio
PIC24FJ64GB004	Microcontrolador
VH 400	Sensor de umidade de solo
DS 1822	Termômetro digital
NRF24LO1	Módulo transmitter e receiver
DHT11	Sensor de temperatura e umidade do ar
NRF24L01	Transmissor de dados via wireless
DS18B20	Sensor de temperatura
ESP8266	Módulo Wifi
SVM	Support Vector Machine
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
TIP 122	Transistor
LBP	Local Binary Patterns
nm	Nanómetro
UNIOESTE	Universidade Estadual do Oeste do Paraná

Sumário

Lista de Figuras	v
Lista de Tabelas	vii
Lista de Abreviaturas e Glossário	vii
Sumário	viii
1 Introdução	1
2 Trabalhos Relacionados	4
2.1 Plantio	4
2.1.1 Técnicas de melhoramento de germinação	7
2.2 Irrigação automática	9
2.3 Controle de ervas daninhas por reconhecimento de imagem	17
2.3.1 Uso de machine learning	21
3 Arquitetura do sistema proposto	31
3.1 Modelagem física do sistema	31
3.2 Sistema de irrigação e Temperatura	36
3.3 Sistema de reconhecimento de ervas daninhas	37
3.4 Comunicação do sistema com a internet	42
3.5 Testes	43
3.5.1 Sistema de Irrigação	43
3.5.2 Sistema de Iluminação	45
3.5.3 Sistema de reconhecimento de ervas daninhas	45
4 Conclusões	49
Referências Bibliográficas	51

Resumo

A agricultura de baixa escala vem gradualmente mostrando ser uma solução saudável por não utilizar pesticidas e financeiramente viável para ser utilizada no dia a dia da população. Porém ainda esbarra em problemas de manutenção dos cultivos plantados. A utilização de métodos de agricultura de precisão podem auxiliar no cultivo residencial de baixa escala. Este trabalho teve como objetivo implementar soluções de software e hardware para a criação de um protótipo autônomo capaz de controlar o ciclo de vida de um cultivo de alface. Uma das funcionalidades desenvolvida é a captura e análise de imagens adquiridas por uma webcam que se desloca pela extensão do cultivo, visando o reconhecimento da possível existência de plantas invasoras. O sistema, através de sensores, monitora o índice de umidade do solo e, quando necessário, realiza a irrigação automática da plantação. O protótipo tem o foco no cultivo em ambientes *indoor*, utilizando fontes de luz artificial para possibilitar o crescimento mesmo sem a incidência de luz natural. A fim de facilitar a interação do usuário o sistema periodicamente envia as informações atuais do cultivo para a aplicação web do twitter, garantindo ao usuário uma simples interação com sua plantação. Os testes realizados com o sistema de detecção utilizando padrões e também a solução em nuvem AutoML da Google apresentaram resultados imprecisos, indicando a necessidade de exploração de outros métodos para realizar o reconhecimento eficiente de plantas invasoras. Desta forma, apesar da arquitetura do sistema fornecer condições para a aquisição de imagens e apesar da segmentação dos objetos presentes na imagem entregar resultados satisfatórios, a fase de rotulação do sistema acabou não funcionando.

Palavras-chave: Agricultura de Precisão, Agricultura de baixa escala, Dispositivos Embarcados, Análise de Imagens

Capítulo 1

Introdução

Segundo estudos, a descoberta da agricultura ocorreu entre dez e doze mil anos atrás, durante a pré-história, no período neolítico (MAZOYER, 1997). Alguns indivíduos, parte de tribos caçadoras coletoras, descobriram que alguns grãos coletados na natureza para sua alimentação poderiam ser semeados, caso fossem enterrados. Após o surgimento da agricultura, as tribos nômades tiveram um aumento na oferta de alimento, possibilitando residir em um lugar fixo e, conseqüentemente, mais seguro. Com os benefícios advindos da descoberta da agricultura, as tribos que antes eram nômades conseguiram evoluir tecnologicamente (DIAMOND, 2002) e em termos populacionais.

Após a descoberta da agricultura, a revolução industrial foi o momento que mais impactou o modo como a humanidade produziu alimentos. No século XVII, mais especificamente na Inglaterra, um conjunto de mudanças permitiu o aumento da produção no campo. Neste momento surgiu o começo da agricultura intensiva que, dentre outras características, usava rotatividade de culturas evitando o desgaste do solo, uso de máquinas como a semeadora mecânica, o arado e a debulha. Com o uso massificado de agricultura intensiva, a quantidade de pessoas necessárias para o trabalho no campo diminuiu substancialmente e a população mundial teve um aumento significativo passando de, aproximadamente, 1 bilhão de pessoas no final do século XIX, para 6 bilhões de pessoas, no começo do século XX (NATIONS, 1999).

Atualmente, a agricultura intensiva é amplamente mecanizada, além de contar com estudos na área da agronomia e engenharia genética, a fim de aumentar e melhorar a qualidade da plantação. Todavia, apesar dos avanços ainda existem problemas, como perda de produção devido à estiagem, chuva em excesso, pragas, falta de adubo e alto consumo de água para irrigação. Para resolver os diversos problemas encontrados na agricultura, a solução passa pelo maior

uso de tecnologia, através da agricultura de precisão, técnica que usa tecnologia da informação para garantir que as culturas e o solo recebam exatamente o que precisam para um melhor desenvolvimento e produtividade, garantindo rentabilidade, sustentabilidade e proteção ao meio ambiente (OVERTON, 2017).

Um movimento que vem se destacando excepcionalmente, a partir do início do século XX, é a agricultura de baixa escala, representada sobretudo pela agricultura urbana, termo este compreendido como um “*conjunto de atividades de produção animal e vegetal exercidas em meio urbano, visto como espaço abrangente que inclui áreas intersticiais não-construídas e superfícies periurbanas*” (MADALENO, 2002).

A agricultura urbana localiza-se nos espaços públicos ou privados, individuais ou coletivos, focando no autoconsumo e aproveitando-se de forma eficiente e sustentável dos recursos e insumos locais, como o solo, água e resíduos sólidos, entre outros (SANTANDREU, 2007). Entende-se como efeitos positivos do cultivo na área urbana, o complemento de orçamento familiar, uma vez que poderão produzir parte dos seus alimentos e conseqüentemente reduzir os seus gastos na sua compra, melhorar a qualidade dos alimentos ingeridos e, presumivelmente, uma maior segurança alimentar (WEID, 2004).

Considerando que os principais problemas enfrentados para o cultivo em uma área residencial são espaço e a necessidades exigidas pela cultura durante o cultivo, alguns projetos tem focado em resolver estes problemas para o usuário final. Um exemplo de solução é o projeto Farmbot (FARMBOT, 2014) que, utilizando um espaço residencial externo reduzido e empregado tecnologias embarcadas, consegue produzir alimentos sem nenhuma interação posterior ao processo de seleção de culturas. Outro projeto focado na agricultura urbana com que conta com o uso de tecnologia aplicada é o Plantário (PLANTARIO, 2014), caracterizado como uma pequena estufa, que gerencia hortas orgânicas dentro do ambiente residencial; o equipamento controla a iluminação, ventilação e irrigação das culturas de maneira automática, possibilitando um ambiente ideal de cultivo.

Tendo em mente que o futuro da agricultura passa pela agricultura de precisão, e considerando que a agricultura de baixa escala residencial possa usufruir dos avanços tecnológicos na área, este trabalho teve como objetivo o desenvolvimento de um sistema embarcado capaz de realizar irrigação automática e usar o reconhecimento de imagens para identificar ervas daninhas

em certas culturas. O principal foco deste trabalho foi aplicar soluções de software e hardware para a automação das tarefas de reconhecimento por imagens de plantas invasoras e irrigação, focando em uma melhor eficiência relacionada ao uso da água para irrigação, pois os sensores possibilitaram inferir a quantidade de água necessária para a planta. Em relação ao software de reconhecimento, nesta primeira parte, seu uso foi limitado a reconhecer o nível de infestação na cultura, possibilitando, para trabalhos futuros, um método automático de eliminação das pragas reconhecidas pelo software. Os testes foram realizados em baixa escala, com plantas em uma fileira de aproximadamente 3 vasos, facilitando a montagem do experimento em laboratório e também maior controle da execução do mesmo.

O presente trabalho está organizado da seguinte forma: o Capítulo 2 descreve trabalhos relacionados ao tema abordado. O Capítulo 3 apresenta uma especificação detalhada da arquitetura do sistema proposto, sendo a mesma dividida nos componentes de software e hardware empregados, em seguida descreve os testes e resultados obtidos através do sistema proposto. Por fim o Capítulo 4 finaliza este trabalho com as conclusões finais e trabalhos futuros.

Capítulo 2

Trabalhos Relacionados

Este capítulo descreve trabalhos que apresentam algum mecanismo de automação no processo de plantio, irrigação e controle de ervas daninhas. Alguns dos trabalhos listados tem o foco no cultivo de baixa escala como hortas e outros podem ser aplicados em larga escala. A seção é dividida nas etapas de plantio, irrigação e detecção de ervas daninhas por reconhecimento de imagens.

2.1 Plantio

Na área de plantio automático, o trabalho de (SRINIVASAN; PRABHU, 2016) apresenta o desenvolvimento de um robô móvel capaz de plantar sementes automaticamente, com estrutura de alumínio, visando redução de peso. Sua movimentação é realizada pelo uso de esteiras articuladas, fornecendo maior estabilidade, e a direção se baseia em um magnetômetro que guia o veículo, medindo a direção e sentido através de campos magnéticos. Como fonte de energia, são usadas baterias de chumbo-ácido, que fornece energia para o protótipo. Um sensor ultrassônico foi usado para realizar a leitura de informações do ambiente, e que após interpretadas identificam o fim da plantação e possíveis colisões. O conjunto de sensores e motores é controlado por um microprocessador TM4C123GH6PM ARM Cortex M4F. Um modelo 3D do robô pode ser vista na Figura 2.1.

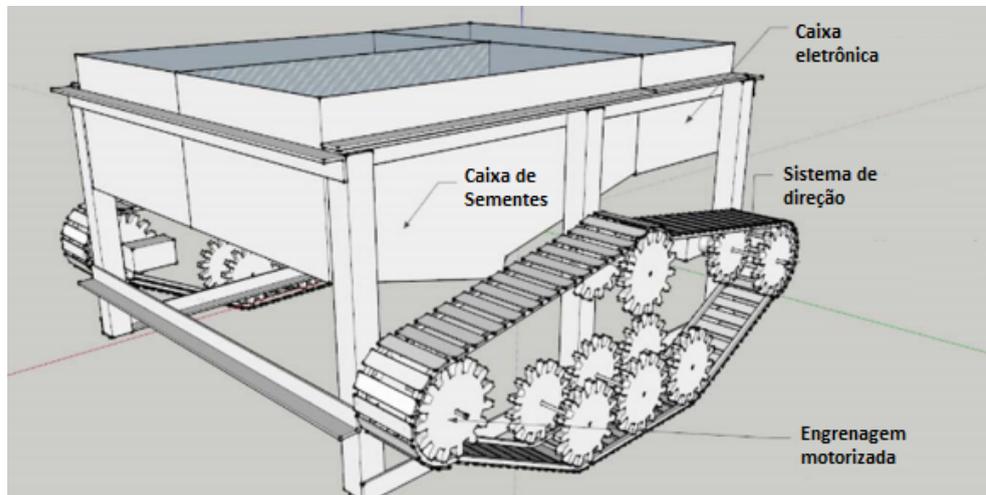


Figura 2.1: Modelagem do robô.
Fonte: (SRINIVASAN and PRABHU, 2016)

O robô é programado para plantar as sementes em linha reta e uniformemente, espaçadas de acordo com uma distância previamente selecionada (varia de acordo com a cultura). Quando é detectado que o robô está na posição correta para a incisão das sementes, um solenoide do tipo *pull – push* é acionado, permitindo a passagem das sementes pelo tubo ligado a caixa de armazenamento, que as conduz para dentro de um funil que direciona o despejo.

Após testes realizados pelos autores, foi observado uma boa taxa de eficiência relacionada a sementes germinadas e um bom nível de automatização do veículo, devido principalmente ao software e aos sensores utilizados. Os testes realizados usaram um solo com ondulações, afim de analisar o desempenho em terrenos acidentados, obtendo-se um resultado satisfatório devido ao uso de um controlador proporcional integral derivativo que faz correções na rota caso ocorra um desvio.

Um outro exemplo de plantio automatizado é proposto pela FarmBot (FARMBOT, 2014), uma *startup* que criou um sistema que realiza e monitora todas as etapas do desenvolvimento de culturas em um ambiente residencial reduzido. O espaço necessário para a implantação de um módulo é de 4.5 m² a 18m². O processo é fisicamente controlado por um braço mecânico, que se movimenta livremente pela área de plantação.

A área de cultivo é reproduzida pelo software do sistema e exibida, via web ou via aplicativo mobile. Após o usuário entrar no sistema, o mesmo pode escolher quais culturas e

em que disposição deseja plantá-las. Na Figura 2.2, observa-se um exemplo de plantio, com as informações das culturas a serem plantadas e suas devidas disposições. O braço robótico retira as sementes de uma caixa de sementes e as planta em posições determinadas, uma vez que o sistema mapeia a área de cultivo e consegue realizar suas atividades com a acurácia necessária.

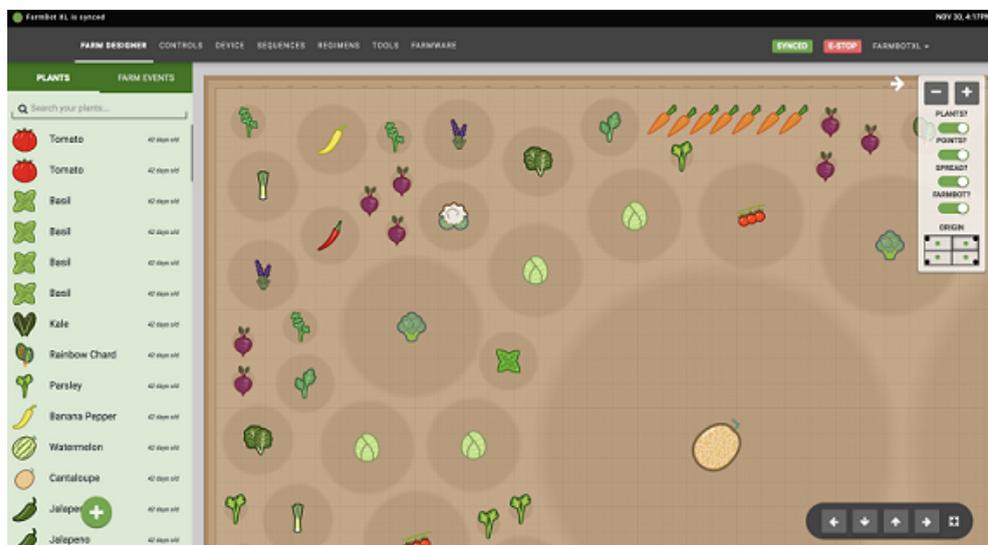


Figura 2.2: Aplicação web para um módulo FarmBot.
Fonte: Farmbot, (2014)

Assim que foi realizado o plantio o sistema, com o auxílio de seu sensor de umidade, verifica o momento correto para a rega, de acordo com a espécie, as condições de solo, o clima e a idade da planta. Todo o processo, segundo os desenvolvedores do projeto é realizado buscando uma grande precisão, afim de que não ocorra nenhum desperdício.

Para a realização do plantio foi utilizado uma peça acoplada ao braço robótico, projetada para minimizar o desperdício de sementes. Seu funcionamento consiste em uma haste fina e oca, conectada a um mecanismo que suga o ar. Através de tal mecanismo, é possível retirar precisamente uma semente da caixa de sementes e inserir a mesma no solo, não se fazendo necessário nenhum outro processo para o plantio da semente. Na figura 2.3 é possível visualizar a peça responsável pelo plantio.

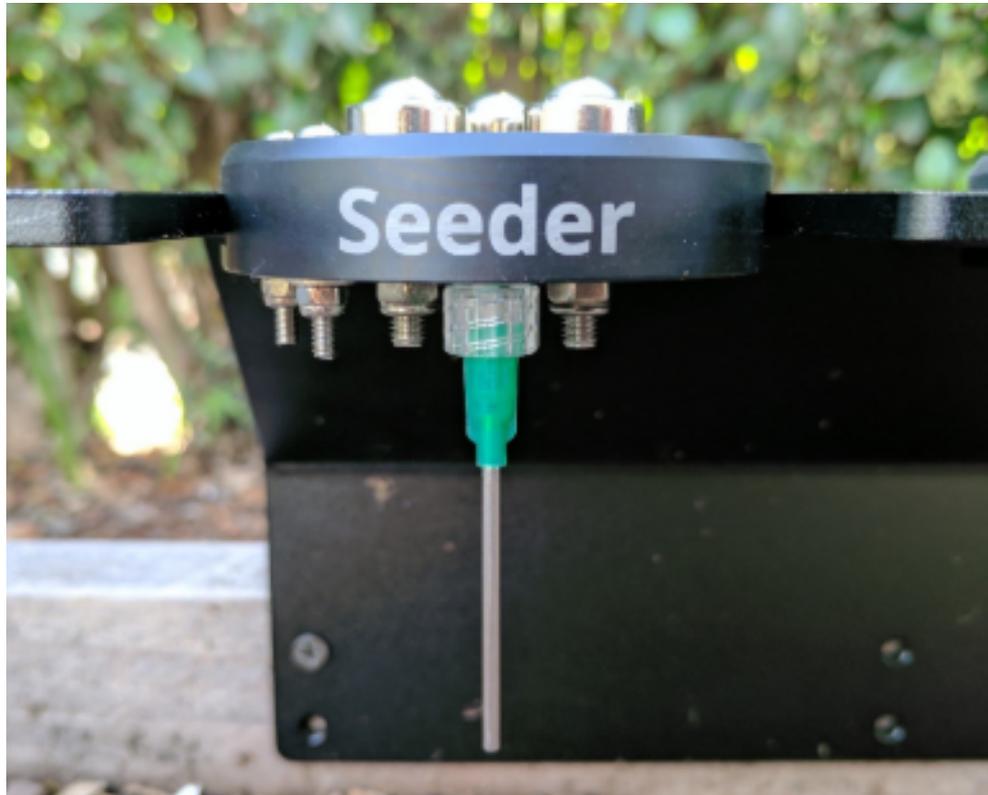


Figura 2.3: Peça responsável pela injeção de sementes.
Fonte: Farmbot, (2014)

2.1.1 Técnicas de melhoramento de germinação

No artigo de (ARVIND; V.ATHIRA, 2017), foi desenvolvido um trabalho na área de germinação e crescimento de plantas. Segundo estudos recentes de (COLLINS, 2001), plantas tem sua taxa de crescimento maior quando expostas a faixas de som audíveis ao ser humano, fator este que também gera benefícios no aumento na saúde de germinação da semente e da transpiração da planta. Um estudo realizado em laboratório foi feito para avaliar os efeitos que tais sons tinham sobre a semente de feijão (*Vigna radiata*), que foi exposta a faixas de 1000 - 1500 Hz, 1500 - 2000 Hz e 2000 - 2500 Hz, com intensidades de 80 dB, 90 dB e 100 dB, durante um período de 72 horas. Na pesquisa avaliou-se o tempo de germinação, o comprimento total e o peso. Os resultados indicaram que as ondas de som tiveram um considerável efeito no tempo de germinação, pois a semente obteve um aumento significativo no crescimento com a aplicação de 90 dB de intensidade e frequência em torno de 2000 HZ. Esse experimento demonstra que cada cultura tem o crescimento aumentado em uma determinada faixa de som. No experimento

os autores optaram pelo uso de uma faixa de som variante, beneficiando um número maior de culturas. Os autores informam no entanto que dados mais específicos sobre outras espécies de plantas não foram estudados e comprovados. A parte de hardware do experimento utilizou-se de dois microcontroladores Arduino, dividindo as funções. O primeiro controla os sensores que obtém informações de temperatura e umidade do solo na área de cultivo, e o segundo Arduino utilizado no projeto controla um emissor de som que gera as frequências necessárias para o crescimento mais acelerado das plantas, válvulas para a liberação de água e *sprinklers* (aspersor) para a irrigação.

Quanto a comunicação dos microcontroladores e o computador principal, um módulo Zigbee (ZIGBEE, 2019) foi utilizado. Após as informações chegarem, um algoritmo é aplicado e toma as decisões de acordo com os dados coletados pelos sensores e são armazenados no banco de dados, que contém os limiares dos sensores e as previsões do tempo. A arquitetura do sistema pode ser melhor analisada na Figura 2.4

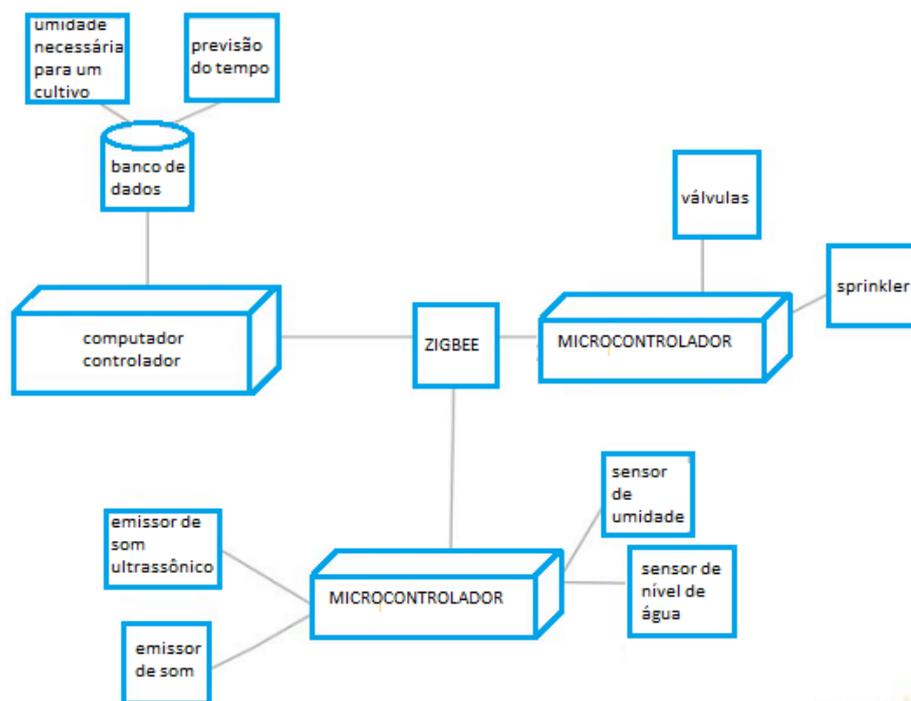


Figura 2.4: Diagrama de Arquitetura.
 Fonte: Adaptada de COLLINS e FOREMAN, (2001)

2.2 Irrigação automática

No trabalho de (GUTIÉRREZ et al., 2014), foi desenvolvido pelos autores, um sistema com o objetivo de automatizar e economizar recursos na irrigação de cultivos. O projeto consiste em dois componentes de hardware principais, as unidades com sensor wireless (WSUs) que coletam e enviam dados para a central chamada de unidade de informação wireless (WIU), que processa e executa as informações recebidas. As WSU são ligados via rádio com a WIU, transmitindo as informações captadas pelos sensores para a central, assim implementando uma rede de sensores sem fio (WSN) que usa a tecnologia ZigBee. A unidade de informação wireless também conta um módulo GPRS para transmitir toda a informação para a internet. A Figura 2.5 apresenta a arquitetura utilizada no trabalho.

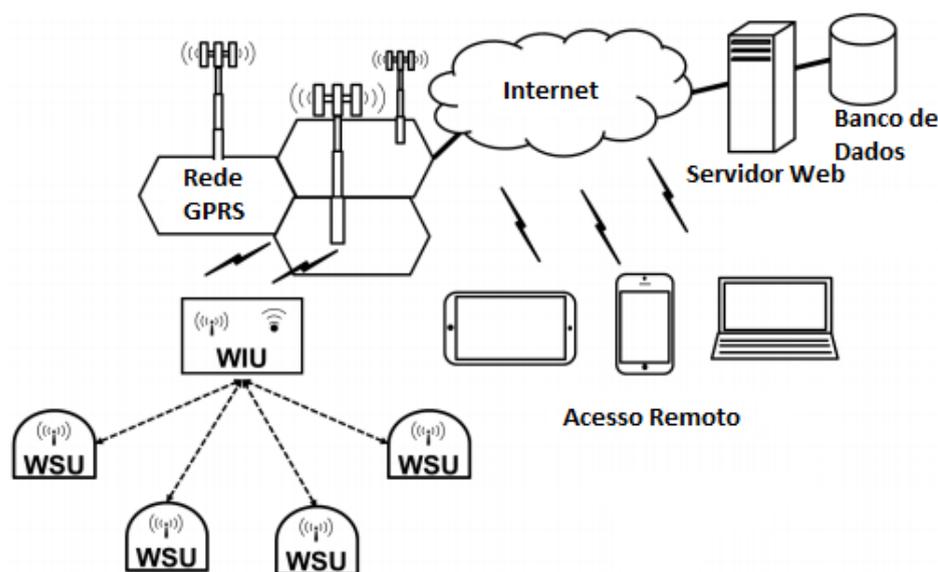


Figura 2.5: Design Sistema.

Fonte: Adaptada de *Gutiérrez et al.*, (2014)

O sistema apresentado na Figura 2.5, tem a grande vantagem de permitir a escalabilidade, pois conforme o tamanho da área de cultivo pode ser selecionado o número necessário de unidades WSU. Tais unidades possuem um microcontrolador de modelo PIC24FJ64GB004, programado em C para controlar os sensores e o módulo ZigBee, uma solução de rede sem fio (WSN) de curto alcance, baixo consumo de energia e baixo custo de aquisição, que se mostrou uma boa opção por possuir maior alcance que opções com *Bluetooth*, conseguindo operar em um raio

de até 1500m em ambientes abertos. O sensor VH 400, conectado a unidade WSU, é utilizado para medir a umidade do solo; o modelo foi escolhido devido ao baixo consumo de energia (< 7 mA). O sensor de temperatura do solo escolhido foi um termômetro digital DS 1822. Para alimentar a central, foram usadas 3 baterias recarregáveis e um pequeno painel solar de tamanho 94 mm x 75 mm com capacidade de 50 mA a 4.8V, que, segundo os autores, é o suficiente para alimentá-la.

Na figura 2.6 é possível ter uma melhor visualização de uma unidade WSU. Mais especificamente, no item A da figura vemos a placa de circuito impresso (PCB), no item B é ilustrado o modem Zigbee, no item C o sensor de temperatura, no item D o sensor de umidade, no item E as baterias do módulo, no item F uma célula fotovoltaica e, por fim, no item G a cápsula protetora do módulo.

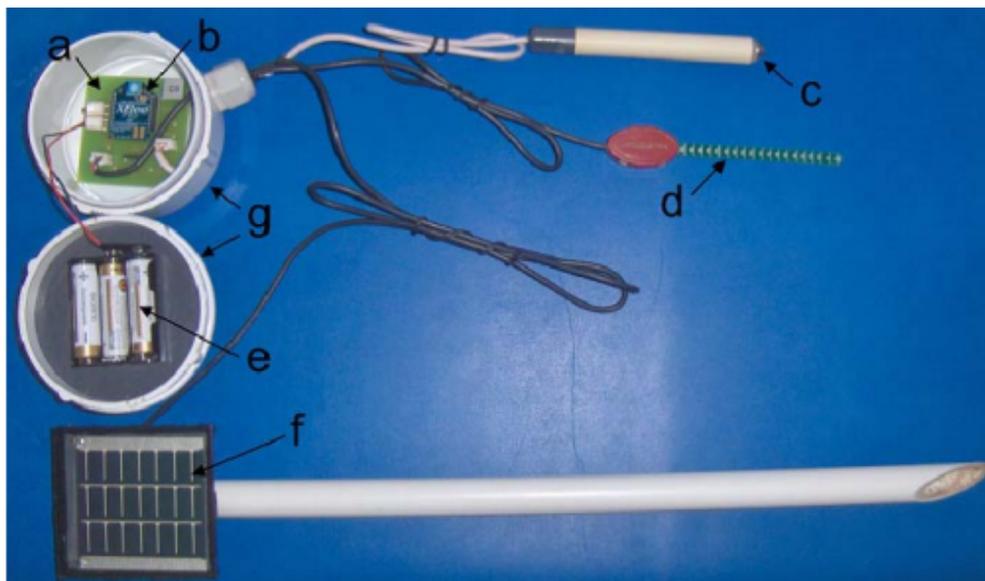


Figura 2.6: Unidade WSU.
Fonte: *Gutiérrez et al.*, (2014)

Como visto na Figura 2.5, existe apenas uma WIU, que recebe, identifica, armazena e analisa as informações enviadas pelas unidades com sensor wireless. Para tais tarefas, os componentes da WIU são: um microcontrolador master de modelo PIC24FJ64GB004 (programado para realizar download de data e hora de um servidor na internet, transmitir para as WSU via Xbee, receber e analisar as informações das WSU e armazenar as dados de tempo de irrigação), um módulo GPRS MTSMC-G2-SP (faz a comunicação com a internet) e, por fim, um módulo

de irrigação que controla duas bombas através de relés eletromagnéticos ligados ao microprocessador principal. As bombas consomem 48 Watts cada e são abastecidas por um tanque de 5000L.

Na Figura 2.7 é possível ter uma melhor visualização da unidade central WIU. Mais especificamente, no item A.1 da figura vemos o PCB, no item B.1 o microcontrolador central, no item C.1 a memória flash, no item D.1 o isolador ótico, no item E.1 uma interface RS-232, no item F.1 um botão responsável por acionar a irrigação de forma manual, no item G.1 Cabos de saída pra as bombas e, no item H.1, o cabo de energia do módulo. Na parte direita da imagem, é mostrado em A.2 um modem Zigbee, em B.2 o modulo GPRS, em C.2 um cartão SIM para a conexão com a internet, em D.2 uma antena GPRS e, finalmente, em E.2 os relés das bombas

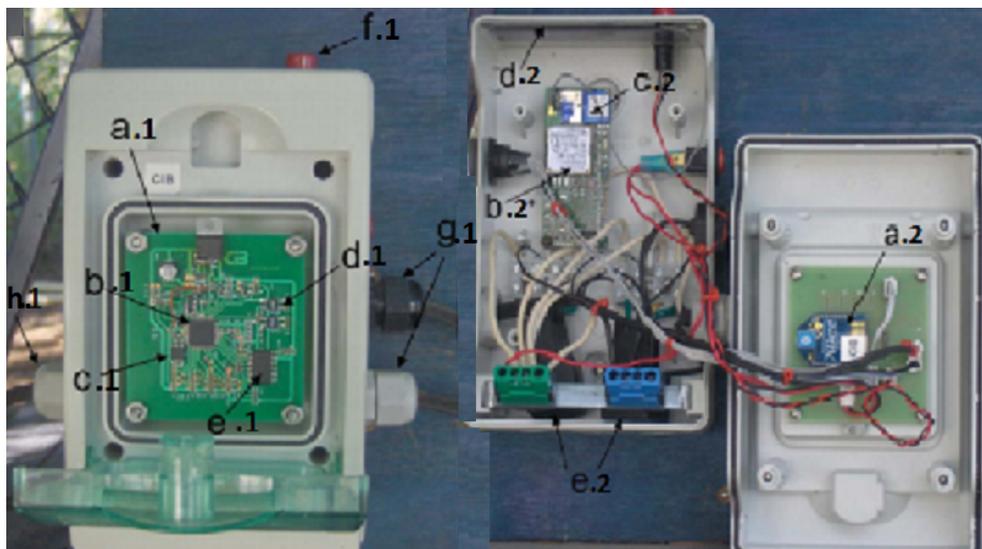


Figura 2.7: Unidade central WIU.

Fonte: *Gutiérrez et al.*, (2014)

Através da aplicação web desenvolvida no âmbito do projeto é possível escolher dentre quatro opções de irrigação, sendo elas:

- Duração fixa para irrigação manual, através do botão localizado na WIU;
- Irrigação com data e hora programadas através da página web;
- Irrigação automatizada com duração fixa, ativada se pelo menos um sensor de umidade do solo tem um valor menor que o limite mínimo, predeterminado de acordo com o cultivo

analisado;

- Irrigação automatizada com duração fixa, ativada se pelo menos um sensor de umidade do solo tem um valor menor que o limite mínimo, predeterminado de acordo com o cultivo analisado;

No trabalho proposto por (RAO; SRIDHAR, 2018) , é apresentado um sistema de irrigação IoT, utilizando sensores de temperatura, de intensidade de luz, de umidade do solo e de umidade relativa do ar, interconectados em uma plataforma Arduino, além de um NRF24LO1 *transmitter* e *receiver* (NORDIC SEMICONDUCTOR, 2006), responsável pela comunicação wireless, com a aplicação web que analisa as informações, armazena os dados e toma decisões. Na Figura 2.8, é possível visualizar a arquitetura do sistema proposto.

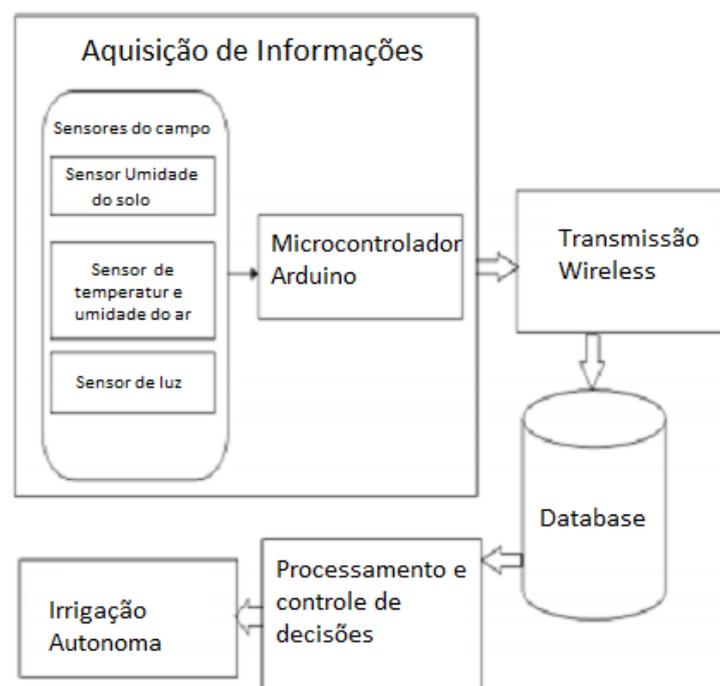


Figura 2.8: Design Sistema.
Fonte: Adaptado de *Rao e Sridhar*, (2018)

A primeira etapa da aplicação é a aquisição de informações, realizada pelos sensores distribuídos na área de plantação. Cada conjunto de sensores é protegido por uma caixa destinada

ao armazenamento dos dispositivos. Cada sensor extrai um tipo de dado, aumentando as informações obtidas do ambiente. Um sensor dielétrico coleta dados relacionados a umidade do solo. Já o sensor de modelo DHT11 extrai dados de temperatura e umidade do ar; a medição é realizada antes e depois da irrigação, pois as informações são alteradas após a umidade do ar aumentar, porém vale ressaltar que o sensor não é a prova d'água, devendo ficar fora do alcance dos *sprinklers*. Por fim, um sensor de luz, no caso um resistor dependente de luz (LDR), tem a tarefa de detectar o nível de intensidade de luz ambiente, tendo uso específico para estufas com auxílio de iluminação artificial, assim quando é detectado um valor baixo de luz, a iluminação secundária é acionada. A funcionalidade de controle de luz foi implementada no trabalho de maneira opcional, dependendo da necessidade do usuário. Na fase de transmissão de dados via wireless, os autores utilizam um módulo NRF24L01, em que a transmissão é feita entre as unidades dispostas no campo (*transmitter*) e um único *web server* (*receiver*). O módulo usa um transceptor de 2.4 GHz, com uma taxa de transmissão de 256 kbps/s. Dependendo da extensão do campo, pode haver vários transmissores, mas um único *receiver*, conectado a um *web server* via conexão Ethernet. A última etapa do sistema proposto pelos autores é o armazenamento das informações recebidas em um banco de dados mysql, processamento das informações adquiridas e realização das tomadas de decisões com as mesmas. Apresenta-se o fluxograma do sistema na Figura 2.9.

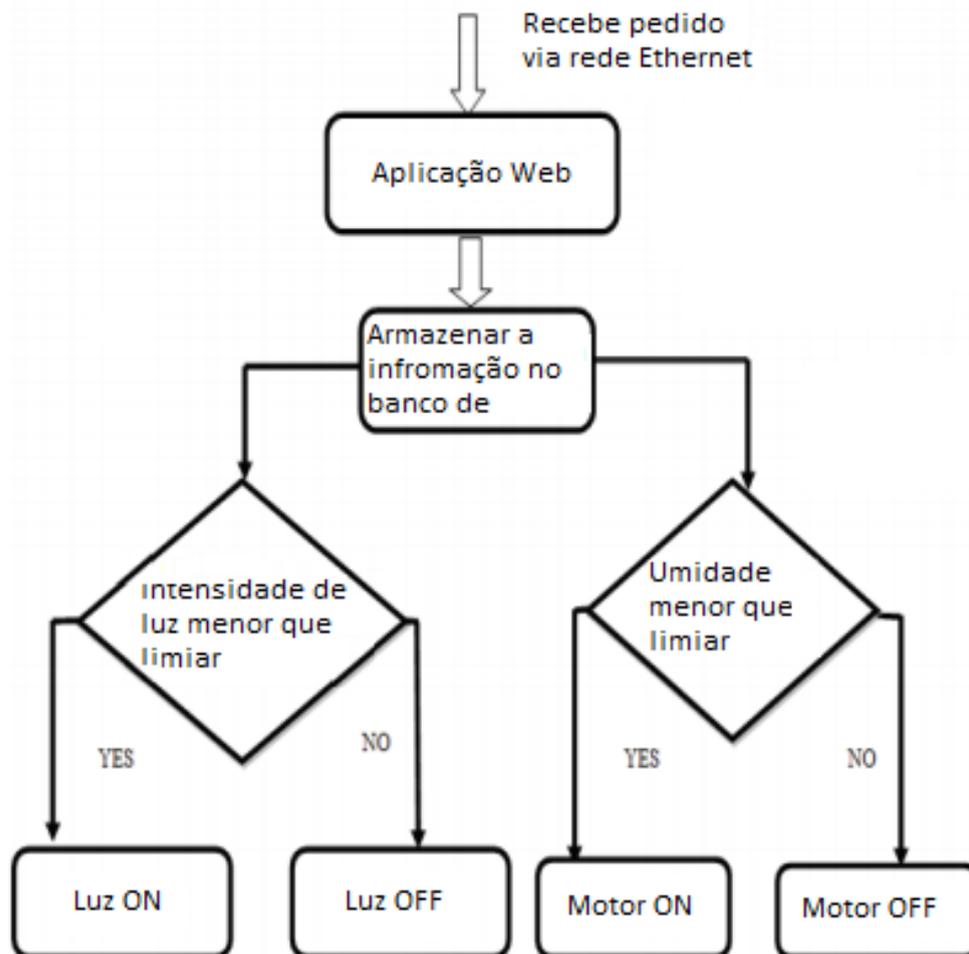


Figura 2.9: Fluxograma do sistema.
 Fonte: Adaptado de *Rao e Sridhar*, (2018)

A parte de processamento de informações verifica os vários dados recebidos pelos sensores e, com os valores de limite já definidos, realizar a atuação na luz e irrigação. Vale mencionar que os valores mudam de acordo com a cultura plantada, pois cada uma tem suas necessidades específicas. Ainda que, os valores são alterados de acordo com as condições climáticas, sendo que a umidade do solo muda de acordo com as estações, ou seja, os valores limites são alterados de acordo com a época do ano, para a realização o sistema consulta informações de data, descobrindo assim quais limiares utilizar. Após as verificações, caso for identificado valores abaixo do limite, os relés ou as luzes são ligados para a irrigação ou iluminação, respectivamente. Todavia, caso o usuário deseje fazer a irrigação de forma manual, é possível via a aplicação web ou o app mobile desenvolvido para facilitar o uso.

No trabalho apresentado por (SINGH; SAIKIA, 2016), foi observado que durante o processo de irrigação, pode ser difícil mensurar a quantidade de água usada. Por esse motivo, aplicou-se uma abordagem que considera esta informação relevante, permitindo a comparação de dados com períodos passados ou uma análise de quantidade de água usada durante certo período. Os autores do projeto, cujo objetivo era desenvolver um sistema de irrigação inteligente usando Arduino, optaram por utilizar no projeto um sensor de fluxo de água, cujo funcionamento se dá por uma válvula em sua estrutura, com formato de catavento e um ímã acoplado que trabalha em conjunto com um sensor de efeito hall, encarregado de enviar um sinal PWM. Através destes pulsos é possível mensurar a vazão de água. Na Figura 2.10, o sensor de vazão de água pode ser visualizado.



Figura 2.10: Sensor vazão de água.
Fonte: *Singh e Saikia, (2016)*

No mesmo trabalho foi usado um sensor de temperatura DS18B20, a fim de monitorar a temperatura na plantação, sendo uma informação relevante pelo fato de a temperatura da planta

ser de suma importância para o seu desenvolvimento. Em períodos de alta temperatura pode ocorrer desidratação e, conseqüentemente, a malformação ou morte da planta. Sendo assim, em períodos de alta temperatura, o sistema aplica irrigação com o intuito de diminuir a temperatura da planta, melhorando a saúde da mesma. Além dos dois sensores apresentados, utilizou-se um sensor de umidade de solo, ligado a um microprocessador Arduino e a um módulo Wifi ESP8266, que se comunicam com um web server responsável pela análise de dados e tomada de decisão. Ao analisar a imagem da Figura 2.11, podemos ter uma visualização de como o sistema funciona, possibilitando optar pela escolha da irrigação manual via software, disponibilizando os dados atuais enviados pelos sensores e exibindo uma tabela com os valores ideais de cada cultura. Por exemplo, o item 1 da Figura 2.11 que representa a cultura do arroz, a terceira coluna indica o nível de umidade atual dentro de uma escala feita pelos autores do sistema, a quarta coluna exibe a quantidade de água medida em milímetros que o cultivo necessita durante seu desenvolvimento, por fim na quinta coluna é exibido a temperatura ideal para o cultivo do arroz.

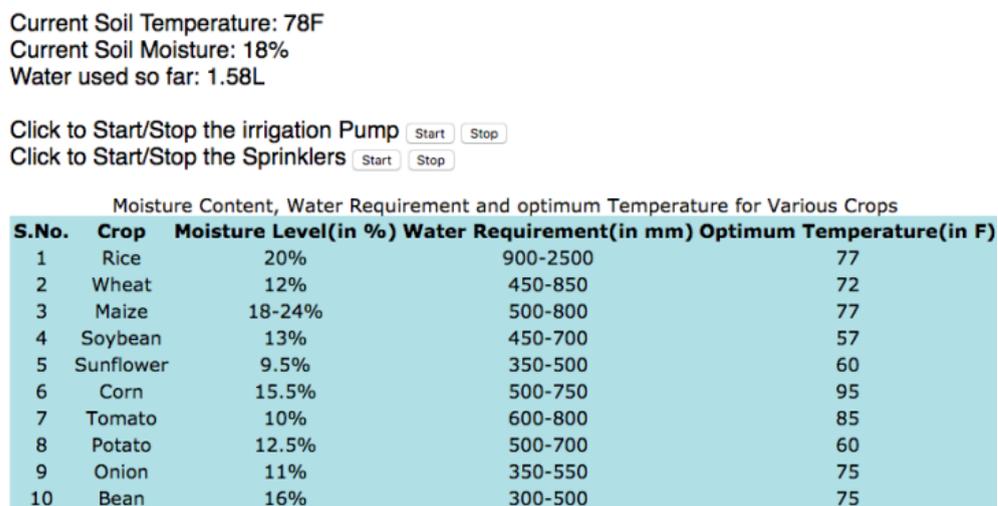


Figura 2.11: Webpage do sistema.
 Fonte: *Singh e Saikia, (2016)*

Apesar do sistema proposto ser de simples implementação, foram usadas abordagens interessantes que possibilitam um melhor cuidado com a planta e redução do consumo de água a um baixo custo energético. Segundo os cálculos feitos pelos autores, em um período de dois meses, em uma área de 600 m² foram gastos 341 m³ de água através do método tradicional de irrigação. Já com o sistema proposto, o gasto foi de 29 m³. Relacionado ao consumo de ener-

gia, constatou-se conseqüentemente uma redução de energia, uma vez que os sprinklers foram acionados por menos tempo.

2.3 Controle de ervas daninhas por reconhecimento de imagem

As características atribuídas as ervas daninhas são fácil sobrevivência, alto nível de propagação, fácil adaptação e competição com a cultura por água, luz, nutrientes e dióxido de carbono, sendo este o principal motivo do interesse por sua eliminação. Nas plantações, a presença de ervas daninhas é considerada um problema, contornado por uso de defensivos agrícolas em escala comercial, desta forma alguns trabalhos estão sendo desenvolvidos para diminuir a quantidade de agrotóxicos usada nos cultivos. Uma das possibilidades é usar processamento digital de imagens, para reconhecer as ervas daninhas de uma determinada cultura. Assim, após o algoritmo realizar o reconhecimento, o controle ativo empregado realiza uma eliminação seletiva, e conseqüentemente mais eficiente. No trabalho escrito por (SANTOS, 2009), houve um estudo acerca de reconhecimento de padrões de plantas invasoras. O trabalho descreve como foram feitos os descritores de imagens para o reconhecimento de padrões de plantas invasoras. O processo se dá pela aquisição de imagens, pré-processamento, segmentação por textura e descritores de formato. O sistema de reconhecimento desenvolvido é capaz de reconhecer plantas invasoras de folhas largas e estreitas. A Figura 2.12 apresenta o diagrama de blocos referente ao sistema.

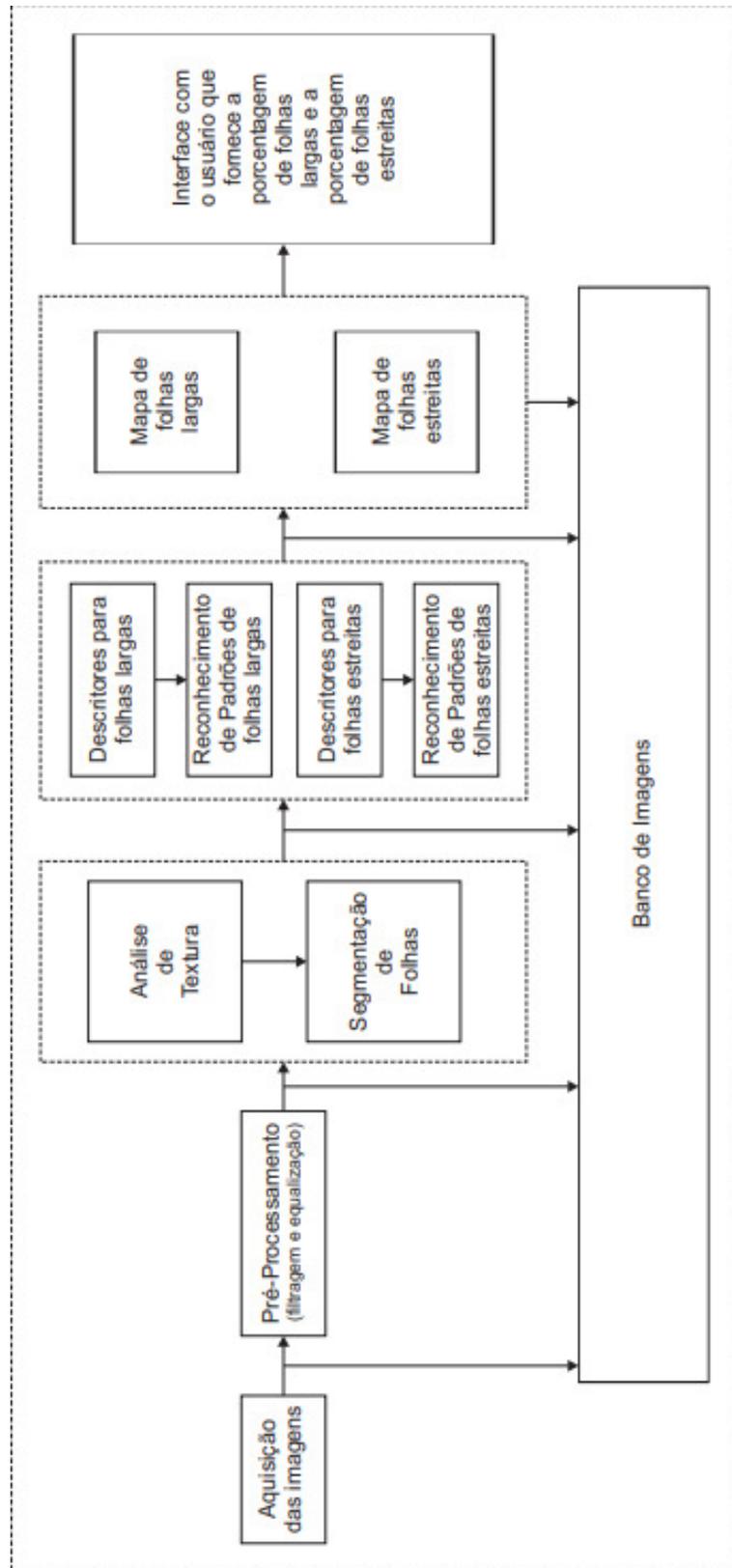


Figura 2.12: Diagrama de blocos do sistema.
 Fonte: SANTOS, (2019)

A aquisição das imagens é feita por uma câmera digital de uso comercial, que captura a cultura que será analisada e seu entorno. Com a imagem já adquirida, tem-se início a etapa de segmentação, que tem, em sua primeira fase, a aplicação dos métodos de pré-processamento. Tais métodos foram divididos em cinco etapas, sendo elas:

- Etapa 1: A imagem original é dividida em três canais de cores R, G e B.
- Etapa 2: Afim de uma segmentação inicial, é subtraído o componente de cor R do canal de cor G. Uma vez que o canal de cor G representa cores de tom verde tal distinção é feita, visando segmentar objetos deste tom na imagem.
- Etapa 3: A partir da nova imagem, é aplicado um filtro de mediana, retirando ruídos e uniformizando a imagem, sem perder detalhes de bordas, uma vez que tais detalhes são de suma importância para a detecção dos objetos.
- Etapa 4: Na imagem do passo anterior, é feita uma limiarização.
- Etapa 5: É feita uma operação de máscara da imagem resultante da etapa 4 com a imagem original, gerando uma imagem que elimina as informações de fundo, e mantém as informações relativas as folhas.

Após essa etapa, a imagem resultante já consegue distinguir as plantas do fundo da imagem, sendo elas tanto a cultura analisada quanto as ervas daninhas. O próximo processo realizado foi uma segmentação por textura, na qual aplicou-se o método de matriz de co-ocorrência (SONKA; BOYLE, 2013), para obter um mapa de textura das diversas plantas obtidas ainda no pré-processamento. Ao final, o mapa de texturas é calculado e seleciona-se, através dos padrões de interesse, apenas as ervas daninhas presentes na imagem, descartando as plantas que fazem parte da cultura analisada. O processo de segmentação por textura é descrito na Figura 2.13.

Algoritmo 3 Algoritmo para a segmentação baseada em textura

- 1: Obter o mapa de textura:
 - a) Fazer um janelamento na imagem original;
 - b) Calcular o valor de textura de cada janela;
 - c) Normalizar os valores de textura;
 - d) Redistribuir os valores normalizados na paleta de cinza (0-255) para a visualização do mapa;
 - e) Conectar as classes iguais ou dentro de um limite;
- 2: Encontrar na imagem vinda do pré-processamento apenas as regiões de interesse.
- 3: Calcular o mapa de textura das folhas:
 - a) Unir o mapa de textura e a imagem das regiões;
 - b) Calcular para cada região o valor de textura médio.
- 4: Selecionar no mapa de textura das folhas apenas as regiões de interesse.

Figura 2.13: Pseudocódigo segmentação baseado em textura.
Fonte: SANTOS, (2019)

Após a segmentação são usados descritores de imagens, que retiram informações individuais de cada objeto restante na imagem. Com as medidas calculadas podemos comparar com as informações dos objetos previamente conhecidos. No trabalho analisado foram usados descritores baseados na forma das folhas, mais especificamente, descritores de redondez, alongação e solidez. Com os dados colhidos pelos descritores, faz-se o uso do pseudo-algoritmo apresentado na Figura 2.14, que faz o reconhecimento das regiões encontradas e realiza uma comparação entre o valor obtido para cada região e a média de cada um dos padrões. O objeto é classificado como folha larga caso a distância do seu valor seja menor para a média de folhas largas do que para a média de folhas estreitas ou classificado como folha estreita, caso o algoritmo conclua o contrário.

Algoritmo 4 Algoritmo para o reconhecimento das folhas

- 1: Encontrar a média do descritor para cada um dos padrões (folhas largas e estreitas).
 - 2: Para cada região:
 - a) Calcular o valor do descritor;
 - b) Calcular a distância da região para a média de folhas largas;
 - c) Calcular a distância da região para a média de folhas estreitas;
 - d) Atribuir a região à classe que apresentar a menor distância.
-

Figura 2.14: Pseudocódigo Algoritmo reconhecimento folhas.
Fonte: SANTOS, (2019)

Como apresentado, o sistema baseou-se nas medidas de densidade das plantas invasoras, para classificar as mesmas em folhas largas ou folhas estreitas, afim de fazer a aplicação seletiva de herbicida em tempo real, uma vez que cada tipo demanda uma abordagem diferente. Relacionado ao desempenho, após diversos testes feitos pela autora, chegou-se a uma taxa de acerto do sistema de 84.24% para folhas largas e 80.17% para folhas estreitas. Caso esse sistema seja integrado a outra tecnologia complementar de controle ativo, resultará em uma abordagem eficiente na eliminação de ervas daninhas. Por fim, na Figura 2.15 é possível visualizar o passo a passo do sistema descrito acima.

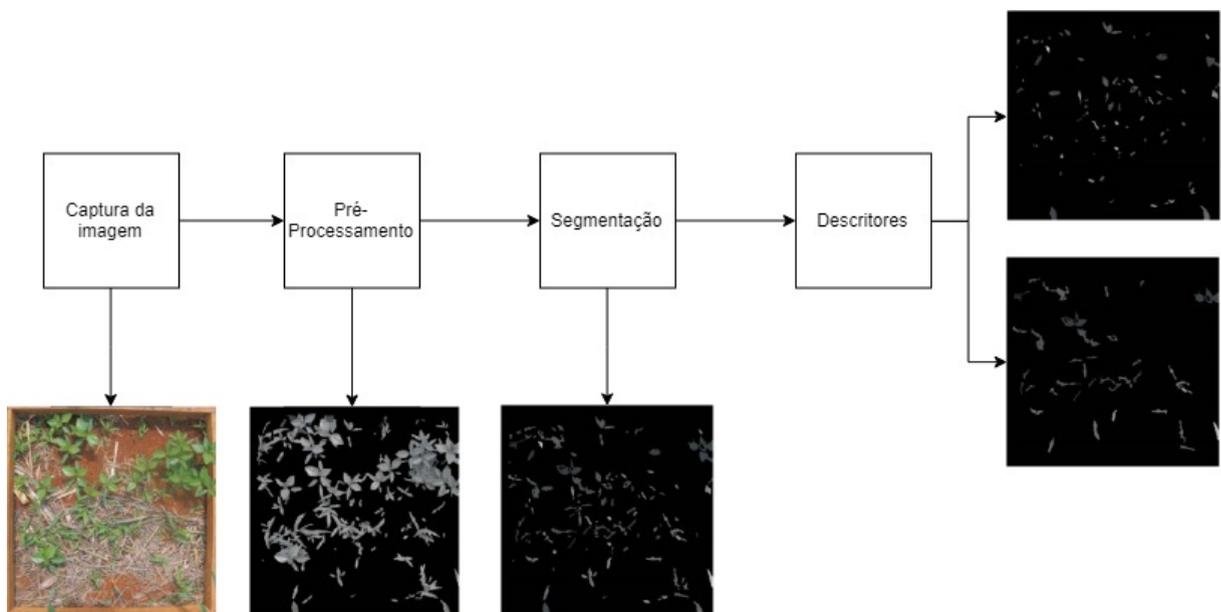


Figura 2.15: Processo de classificação de ervas daninhas..
Fonte: SANTOS, (2019)

2.3.1 Uso de machine learning

No trabalho apresentado por (LAMESKI et al., 2017), foi descrito uma proposta de arquitetura baseada em nuvem, para reconhecimento automático de ervas daninhas. O principal diferencial da proposta é centralizar o processo computacional e disponibilizar o serviço para fazendeiros que ficariam responsáveis por adquirir a imagem, ajudar na detecção semiautônoma na fase de treinamento do software e decidir qual método de eliminação aplicar nas plantas invasoras.

O primeiro obstáculo no desenvolvimento seria a detecção de diferentes tipos de plantas in-

vasoras, estas adquiridas por diferentes sensores de captura de imagem, pois parte de aquisição de imagens é responsabilidade dos diversos usuários do sistema. Além dos problemas encontrados na parte inicial do processo, também houve dificuldades quanto a solução escolhida pelo usuário para a remoção das plantas invasoras, uma vez que o método escolhido pode ser aplicar herbicidas na área com infestação ou realizar o controle mecânico, sendo que o segundo enfrenta o problema de necessitar de ainda mais precisão, para não danificar a própria plantação.

A opção escolhida pelos pesquisadores, para resolver os problemas encontrados foi aplicar uma arquitetura que suporte alto fluxo de dados, através do uso de *machine learning*, mais especificamente *deep learning*. A escolha pelo método de *deep convolutional networks* foi justificada pelos estudos realizados no artigo de (POTENA; NARDI; PRETTO, 2017). Para superar as dificuldades citadas segundo os autores, a arquitetura proposta cobre os seguintes requisitos:

- Lidar com uma grande quantidade de dados, providos por diferentes tipos de fontes;
- Prover informações afim de possibilitar o uso de diferentes opções de hardware para o controle ativo;
- Ser capaz de tomar decisões e dar sugestões para diferentes ambientes e sob condições diferentes;
- Capacidade de processamento rápido;
- Fácil manipulação.

Segundo os autores, o *deep learning* requer muito tempo e esforço para treinamento. Para contornar este obstáculo optou-se em um primeiro momento por um modelo semiautônomo de detecção, deixando o usuário responsável por classificar a planta invasora nesta fase de treinamento. A interface de classificação semiautônoma pode ser melhor vista na Figura 2.16.

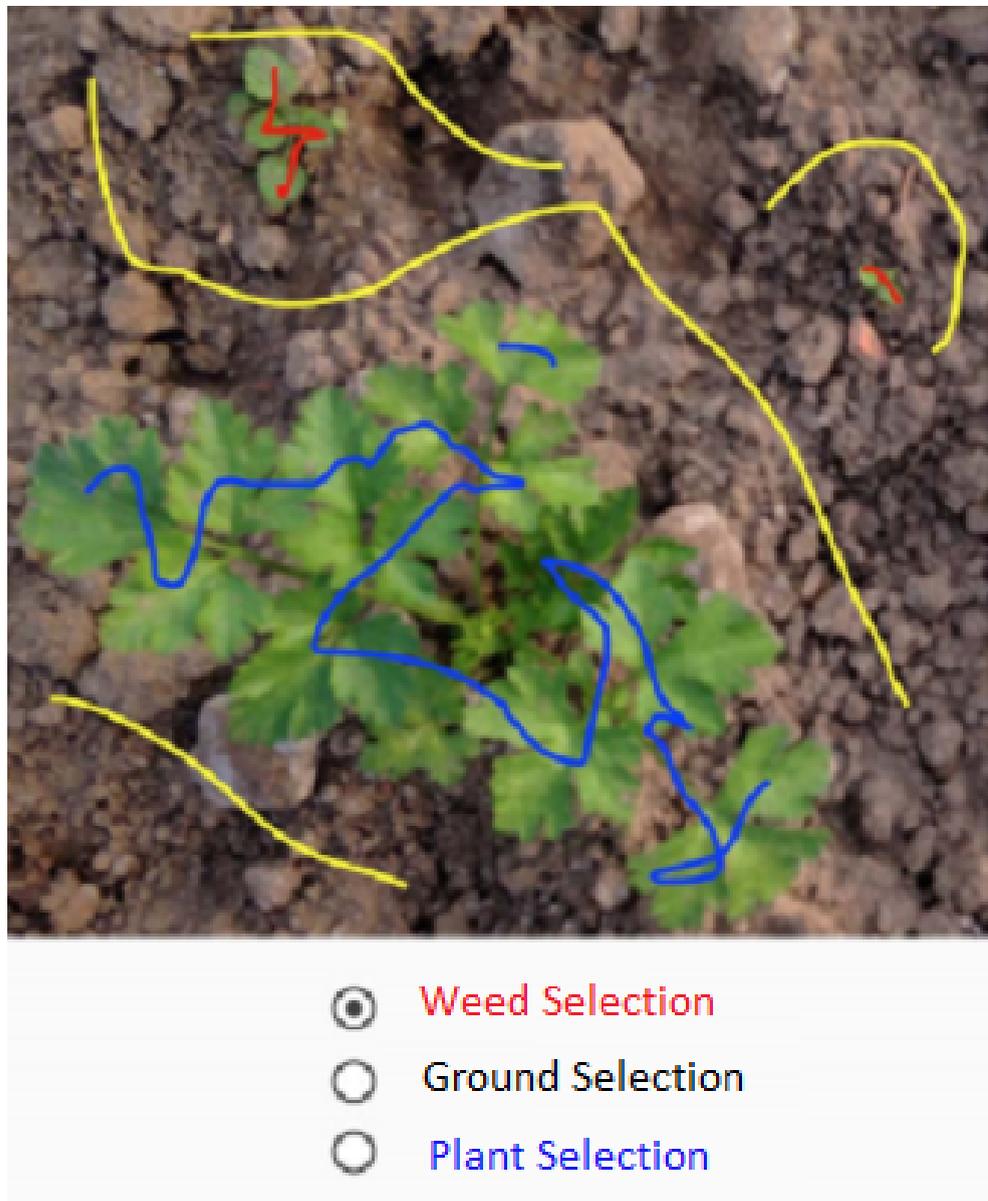


Figura 2.16: Interface de detecção semiautônoma.
Fonte: Lameski et al., (2017)

Para a detecção de maneira semiautônoma é utilizado um algoritmo que diferencia os objetos encontrados na imagem através de um *input* inicial, que designa a área de interesse. A abordagem escolhida foi o *Grab Cut* (KOLMOGOROV; VLADIMIR, 2004), que realiza uma extração de dados da imagem, diferenciando o objeto de interesse, do fundo da imagem. O método implementado pelos autores, garante uma boa eficiência aliada ao fácil uso, quando comparado com outras abordagens. Após a fase de segmentação feita pelo algoritmo *Grab Cut*,

o objeto selecionado é enviado para o treinamento.

A alta complexidade para a solução de *deep learning* foi resolvida com o uso *on-demand* de serviços em nuvem de GPU e armazenamento, de modo que o usuário não precisa ter um hardware para o processamento em suas instalações, sendo necessário apenas o uso de um hardware para adquirir imagens e um sistema mobile, na fase de treinamento.

No trabalho de (SARVINI, 2019) estudou-se quais as melhores abordagens, para o reconhecimento de ervas daninhas em determinadas culturas. Na pesquisa foram feitas comparações entre três abordagens de aprendizado de máquina, *Support Vector Machine* (SVM), *Artificial Neural Network* (ANN) e *Convolutional Neural Network* (CNN). Todas as abordagens foram testadas sobre o mesmo banco de imagens e em seguida tiveram seus resultados comparados.

Na primeira etapa do trabalho foram coletadas as imagens para análise, neste escopo foi capturado imagens de cultivos de Berinjela, Crisântemo, Pimenta, Ragi, Cebola e Nabo. Em relação as plantas invasoras, todas estavam presentes nas plantações analisadas, sendo elas capim angola (*Brachiaria mutica*), tiririca (*Cyperus rotundus*) e Losna branca (*Parthenium hysterophorus*). Ao todo foram capturadas 2560 imagens, estando elas dispostas em imagens apenas com a cultura, apenas com a erva daninha e com ambos presentes. Para a aquisição foram tomados alguns cuidados como tirar as fotos em uma distancia de 50 a 70 centímetros e a aquisição em diferentes ambiente e situações, como solo seco ou úmido, e com diferentes condições de iluminação, com imagens retiradas em diferentes horários do dia. Todo o cuidado com a etapa de aquisição ocorreu devido a necessidade do software reconhecer diferentes tipos de culturas e ervas daninhas em diferentes situações e localizações.

A etapa de pré-processamento, pode ser melhor visualizada na Figura 2.17. Na etapa 1 descrita na imagem, foram aplicados os filtros gaussiano e de mediana para a suavização e retirada de ruídos da imagem. Na etapa 2 foi aplicado uma operação de *resize* na imagem original afim de economizar custo de processamento.

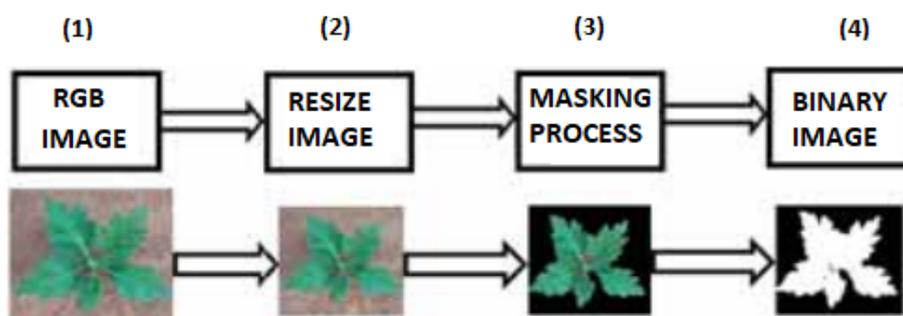


Figura 2.17: Imagem pré-processada.
 Fonte: Sarvini, (2019)

Em seguida na etapa 3, foi extraída a parte verde da imagem através da formula (1), e realizado uma operação de *masking* com a imagem original resultando em uma imagem de fundo preto, e todos os componentes verdes da imagem aparentes. Posteriormente na etapa 4, a imagem sofreu uma limiarização utilizando o algoritmo de Otsu (VALA; BAXI, 2013), que procurou um valor de limiarização t que minimize a variação dentro das classes demonstradas nas formulas (2) e (3) da Figura 2.18.

$$y = 2g - r - b \quad (1)$$

$$\sigma_w^2(t) = W_0(t)\sigma_0^2(t) + W_1(t)\sigma_1^2(t) \quad (2)$$

$$W_0(t) = \sum_{i=0}^{t-1} p(i) , W_1(t) = \sum_{i=0}^{L-1} p(i) \quad (3)$$

Figura 2.18: Formulas usadas no Pré-Processamento
 Fonte: Sarvini, (2019)

Para a extração de características os autores optaram pelas informações de formato da planta. As informações de formato analisadas compreendem área, perímetro, excentricidade, maior eixo e menor eixo, todas as informações destacadas foram retiradas da imagem binária obtida após o pré-processamento.

Por fim, na etapa de classificação de imagens foram aplicados três classificadores que utilizam aprendizado de máquina em sua implementação, cada um deles com suas vantagens e desvantagens, sendo eles:

- 1 - *Support Vector Machine* (SVM): É um modelo de aprendizagem supervisionada que analisa a matriz de características morfológicas de entrada, reconhece os padrões e tenta encontrar um hiperplano ideal, capaz de separar duas classes (KWANG; HANG KIM, 2002). O SVM primeiramente transforma os dados de entrada em um espaço dimensional maior através da função de kernel escolhida, e em seguida constrói um hiperplano ótimo separando as classes, como pode ser visto na Figura 2.19. O SVM é considerado um eficiente algoritmo de aprendizagem de máquina, além de ser recomendado para a realização de tarefas complexas como reconhecimentos de caracteres, voz e imagens. Como desvantagem o SVM pode ser considerado de execução lenta para alguns casos.

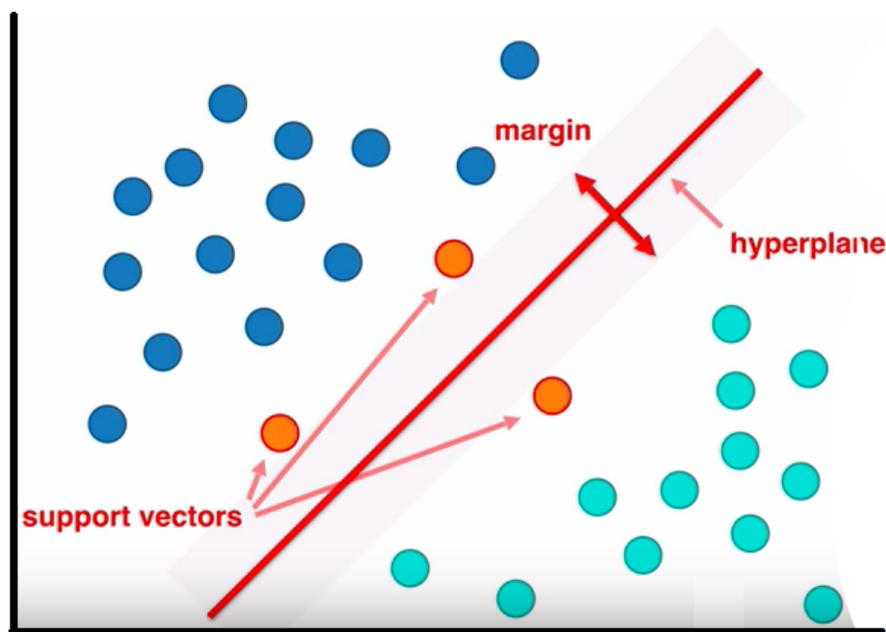


Figura 2.19: Representação Gráfica do Hiperplano.

- 2 - *Artificial Neural Network* (ANN): é um modelo computacional inspirado na estrutura neural de organismos inteligentes que adquirem conhecimento através da experiência. Uma rede neural artificial abstrai e simula o comportamento de uma rede neural biológica.

Na aplicação em questão a camada de entrada da rede neural é fornecida por um vetor com informações de formatos, seguido por uma camada oculta, usada para localizar os recursos proeminentes das classes. Os valores em cada nó são estimados ou aproximados usando uma função não linear. Uma melhor visualização da arquitetura proposta pode ser vista na Figura 2.20.

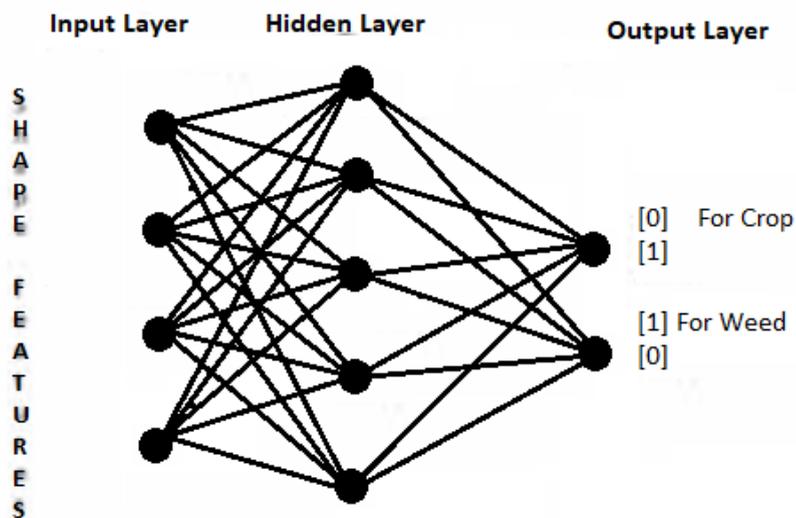


Figura 2.20: Arquitetura da ANN.

- 3 - *Convolutional Neural Network (CNN)*: é uma classe de rede neural artificial, aplicada principalmente para o processamento e análise de imagens digitais. A vantagem de uma CNN é sua capacidade de aprender quais são as regiões de interesse mais relevantes para o conteúdo analisado, desta maneira a quantidade de informações processadas acaba sendo menor *Zhang et al (2015)*. Na abordagem feita pelo autor no trabalho analisado, foram usadas sucessivas camadas convolucionais, com uma função não linear que armazena as características da imagem analisada. A camada totalmente conectada multiplica a entrada por uma matriz com uma função de ativação sigmóide, e em seguida adiciona a um vetor *bias* que contém mapa de atributos. Uma melhor visualização da arquitetura proposta pode ser vista na Figura 2.21

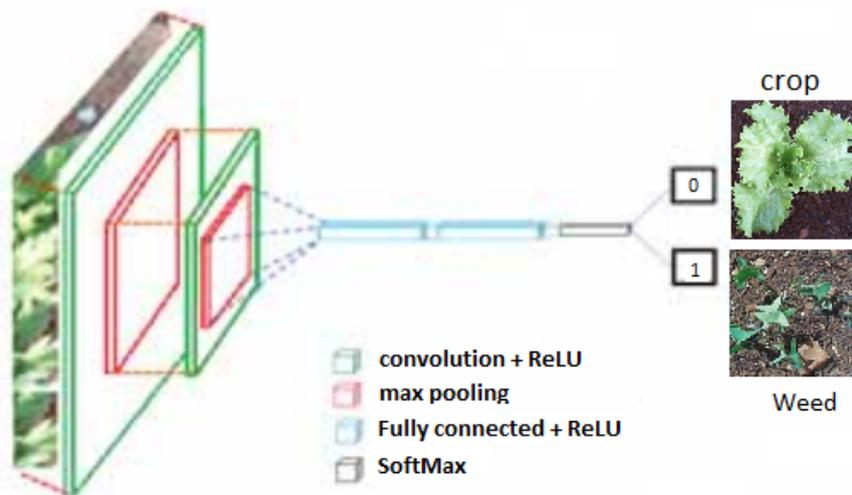


Figura 2.21: Arquitetura da CNN.

Na fase de experimentos, foi usado a linguagem Python (PYTHON, 2019) e a biblioteca OpenCV (OPENCV, 2019). Para o treinamento foi usado uma base de dados com 1155 imagens divididas por suas classes.

- 1 - *Support Vector Machine* (SVM): Foram testados diferentes tamanhos de kernel, no entanto os melhores resultados vieram com um kernel de função polinomial, atingindo 87.6 % de acerto.
- 2 - *Artificial Neural Network* (ANN): Nos testes a rede neural tinha cinco nós de entrada e dois nós de classificação. O número de neurônios na camada escondida variou de 2 a 10. Por fim o melhor resultado ocorreu na combinação de 5-6-2 como podemos ver na tabela 2.1.

Tabela 2.1: Resultados ANN.

Network Architecture	Validation Accuracy(%)	Test Accuracy(%)
5-4-2	95.2	92.8
5-6-2	94.7	93.2
5-8-2	95.4	91.2
5-10-2	95.4	91.6

- 3 - *Convolutional Neural Network* (CNN): Primeiramente foi usado uma camada de entrada de dimensão 250X250X3 executando sobre a arquitetura demonstrada na Figura 2.21. Tal arquitetura foi usada para o treinamento e validação do modelo, aplicando um otimizador a taxa de aprendizado de 0.001 para 10 períodos, com uma base de 231 imagens. Os resultados de validação da CNN pode ser visto na Figura 2.22.

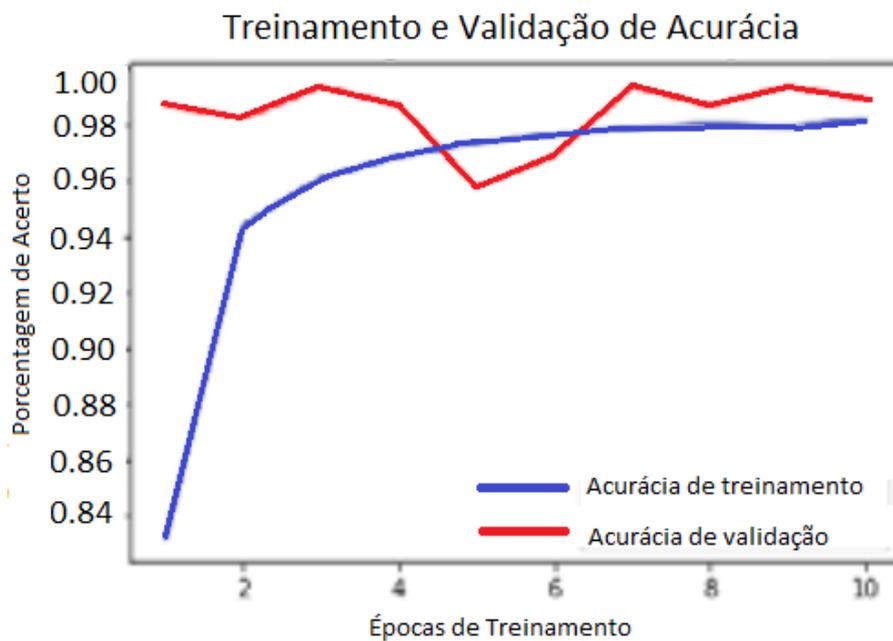


Figura 2.22: Gráfico resultados CNN.

Para a análise dos resultados o autor gerou uma matriz de confusão, afim de ajudar a entender a incerteza dos métodos, em classificar o objeto entre cultivado ou erva daninha. Como analisado na tabela 2.2 as plantas invasoras foram reconhecidas como a cultura analisada mais vezes no SVM e no ANN do que no CNN. Na opinião dos autores, talvez os erros cometidos pelo SVM e no ANN se deram devido a iluminação, ao fundo ou devido a similaridade de padrões entre a cultura e a erva daninha, desta maneira a precisão dos métodos podem ser melhoradas com aplicação de métodos melhores de extração de características. Por fim a conclusão feita pelo autor destaca que o CNN entrega um melhor desempenho devido a sua capacidade de *deep learning* em aprender características relevantes da imagem.

Tabela 2.2: Resultados finais.

	SVM		ANN		CNN	
	Crop	Weed	Crop	Weed	Crop	Weed
Crop	125	0	116	9	125	0
Weed	21	104	11	114	4	121

Capítulo 3

Arquitetura do sistema proposto

Este capítulo descreve a arquitetura do sistema proposto e suas funcionalidades. É detalhado além da modelagem do sistema cada uma das etapas que o compõe. A abordagem utilizada é descentralizada, com o processamento de dados distribuído entre as camadas presentes na arquitetura, podendo ser classificada como uma arquitetura *fog computing* (BANAFÁ, 2014). Este capítulo descreve a modelagem física do sistema, sistema de irrigação, sistema de reconhecimento de ervas daninhas e comunicação do sistema com a Internet.

3.1 Modelagem física do sistema

Para uma primeira análise do projeto, na Figura 3.1 é apresentado uma imagem ilustrativa representando a arquitetura geral do sistema proposto.

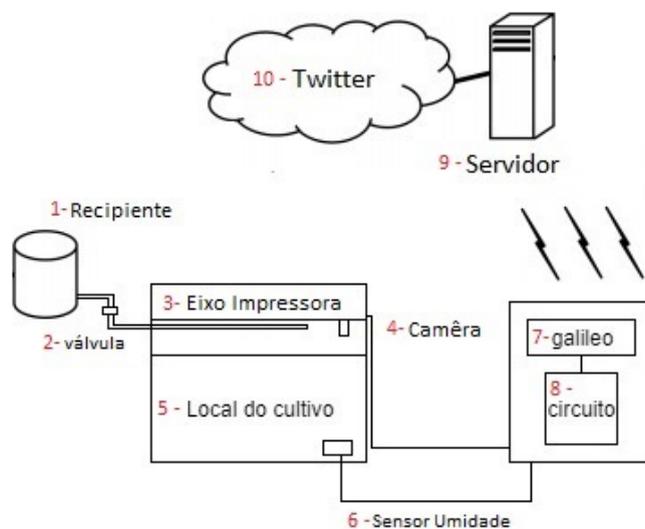


Figura 3.1: Arquitetura do sistema.

Inicialmente, no item 1 da Figura 3.1 é identificado o recipiente onde é armazenada a água utilizada para a irrigação, sua estrutura consiste em um filtro de água adaptado, com sua saída acoplada a uma mangueira ligada até a válvula apontada no item 2 da figura 3.1, acionada pelo sistema quando necessário, possibilitando a vazão de água. Para que a água consiga alcançar toda a extensão do cultivo, a saída da válvula foi conectada a uma mangueira disposta acima das plantas, esta com diversos orifícios espaçadamente alocados em sua parte inferior, caracterizando uma irrigação por gotejamento. A estrutura do recipiente e a válvula podem ser vistas na figura 3.2



Figura 3.2: Estrutura de armazenamento do sistema.

O item 3 da Figura 3.1 consiste em um eixo de uma impressora matricial com 50 centímetros de comprimento, sua função é movimentar uma câmera WebCam fixada onde antes era localizado o cabeçote da impressora, esta que é representada no item 4 da Figura 3.1. A função da câmera é capturar imagens do cultivo, permitindo ao sistema analisar e em seguida retornar ao usuário a porcentagem de ervas daninhas presentes na cultura. A fim de permitir as condições corretas de captura da imagem foi disposta uma lâmpada de LED, ligada apenas no momento da aquisição da imagem afim de entregar boas condições de luminosidade. Além desta fonte de luz é utilizada uma lâmpada de LED específica para o fornecimento de ondas de luz, que possibilitem o crescimento da cultura; esta última lâmpada fica ligada durante um período de 12 horas diárias, mais especificamente das 6:00 as 18:00 horas.

A cultura analisada foi disposta sobre uma caixa de madeira com as dimensões de 50 centímetros de comprimento, 35 centímetros de largura e 15 centímetros de altura, identificada pelo item 4 da Figura 3.1. Nela existe espaço para a criação de três plantas de alface. A terra utilizada no protótipo foi retirada de uma área onde localizava-se um cultivo de hortaliças, a escolha se deu devido a sua reconhecida fertilidade e também devido a mesma possuir uma variedade de ervas daninhas, característica necessária para os testes realizados no projeto. Para a leitura dos níveis de umidade do plantio foi utilizado um sensor de umidade do solo LM393 representado pelo item 6 da Figura 3.1.

Para o controle de todos os equipamentos de hardware utilizados no sistema foi feito uso de um microcontrolador Galileo Gen2 da Intel aliado a um circuito eletrônico montado sobre uma *proto-board*, estes são respectivamente representados pelos itens 7 e 8 da Figura 3.1. O motor de passo utilizado no projeto consiste em um motor bipolar de 4 fios, a partir deste modelo foi montado um circuito utilizando transistores *TIP 122*, resistores de 220 Ohms e diodos, para cada fio de entrada do motor (bobina), é ligado a uma porta digital do microcontrolador, via código as portas são acionadas afim de rotacionar o eixo do motor na direção desejada. Para o acionamento de ambas as lâmpadas e a válvula foram utilizados relés, comandados via Galileo. Para a leitura das informações capturadas pelo sensor de umidade é utilizado uma porta analógica da Galileo. O circuito eletrônico descrito, é apresentado na Figura 3.3.

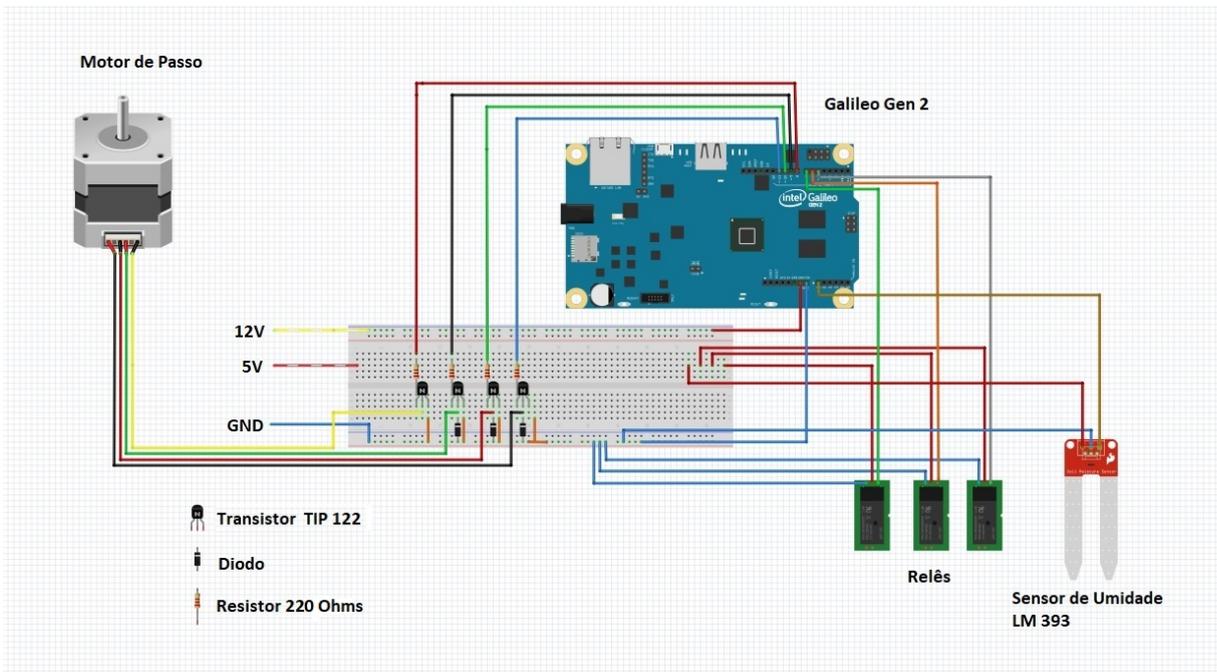


Figura 3.3: Circuito para o controle motor de passo.

O monitoramento e controle do funcionamento do sistema, tal como supervisão e acionamento das lâmpadas, movimentação do motor de passo, aquisição de imagens, leitura e análise das informações do sensor de umidade, acionamento da válvula, entre outros, é realizado através do código embarcado na Galileo. Todavia por uma questão de poder de processamento, após feita a aquisição das imagens as mesmas são enviadas para um servidor que as processa e envia os resultados para o Twitter. Tanto o servidor quanto a aplicação web do Twitter são destacados respectivamente pelos itens 9 e 10 da Figura 3.1.

Para uma melhor visualização do sistema é possível observar a figura 3.4 com o protótipo montado, já a figura 3.5 exhibe como estão dispostos o circuito e o microcontrolador do sistema.



Figura 3.4: Imagem do sistema.

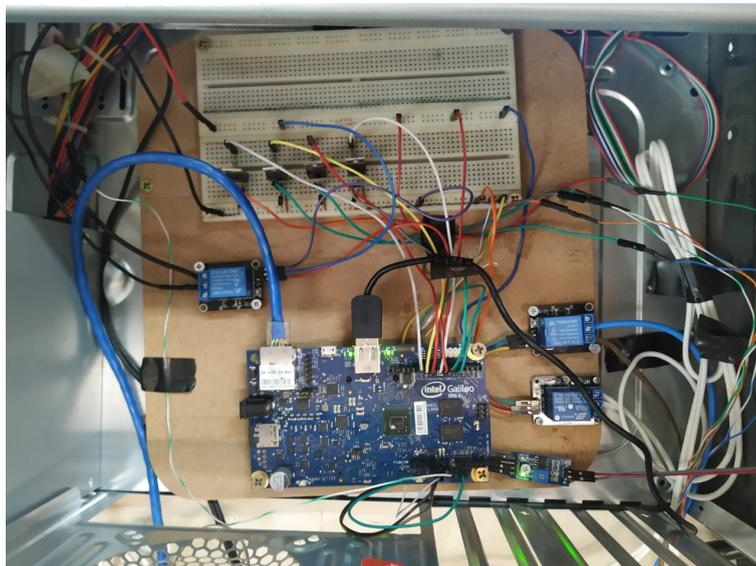


Figura 3.5: Circuito do sistema.

3.2 Sistema de irrigação e Temperatura

Para a irrigação do sistema foi usado um sensor de umidade como o da Figura 3.6. Este equipamento funciona medindo a resistência elétrica entre suas duas hastes paralelas, deste modo quanto maior a umidade de água do solo, menor a resistência (como a água é um dielétrico

sua presença facilita a passagem de corrente). A escala fornecida vai de 400 com o sensor imerso em água, até 900 com o sensor tendo o ar como condutor elétrico. O sensor foi programado para realizar uma leitura na umidade do solo durante intervalos fixos de 20 minutos, caso o valor seja maior que o limiar configurado, um comando é enviado para a válvula abrir durante um período de tempo obtido através de experimentos, no caso um intervalo de 70 segundos.

O limiar adotado durante o período de testes, foi de 600, tal valor foi alcançado por meio de testes empíricos relacionados a umidade do solo. Quanto a temperatura, durante o período que o sistema foi testado o ambiente indoor de cultivo, foi mantido a uma temperatura de 25 graus com o auxílio de um ar condicionado, a temperatura foi escolhida após pesquisas relacionadas a condições ideais de crescimento do cultivo (ANDRIOLO, 2003).

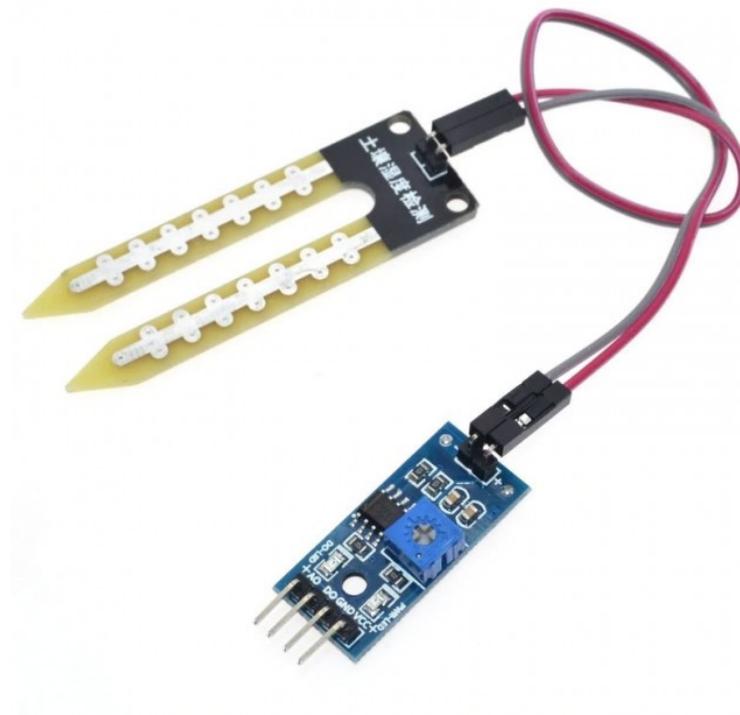


Figura 3.6: Sensor de Umidade.

3.3 Sistema de reconhecimento de ervas daninhas

Neste módulo do sistema é implementado um Sistema de Visão computacional, desenvolvido para identificar a porcentagem de ervas daninhas na imagem, em relação a cultura anali-

sada. Na Figura 3.7 é possível visualizar todos os passos implementados neste módulo.

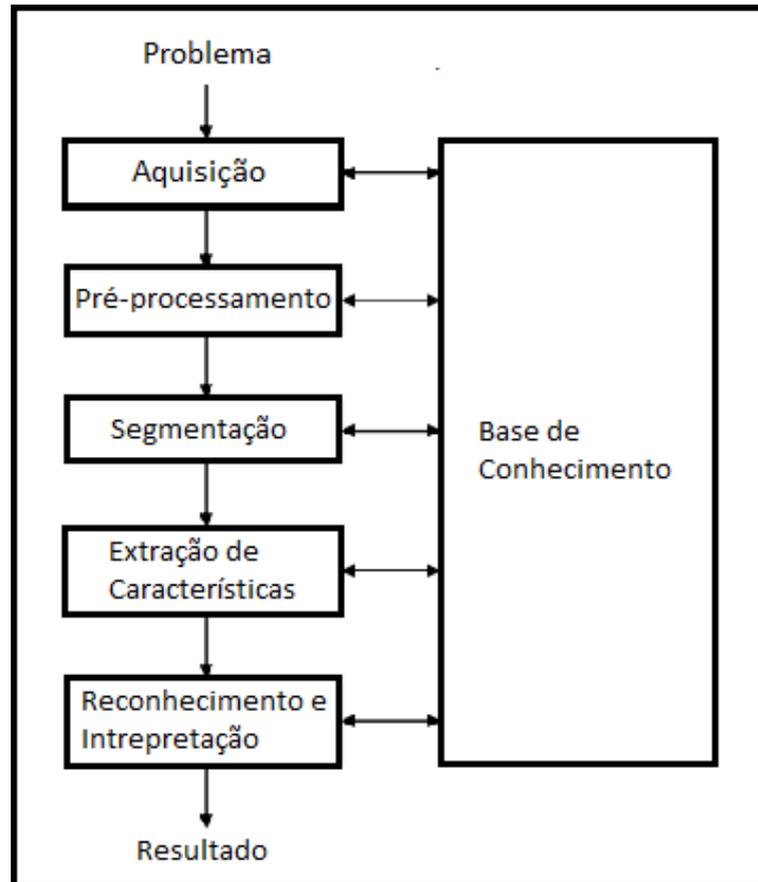


Figura 3.7: Sistema de Visão computacional (SVC) e suas principais etapas.

- **Pré-Processamento:** Nesta etapa é criada uma nova imagem com apenas as folhas presentes na imagem original, não diferindo as folhas da cultura com as folhas das plantas invasoras. Portanto nessa nova imagem formada é eliminado todas as informações de fundo da imagem. Para chegar a imagem mencionada foi aplicado os seguintes processos:

1 - A imagem original RGB é convertida para o formato HSV, em seguida é aplicada uma segmentação de cor mantendo na imagem apenas as partes correspondentes ao espectro de cor verde. Neste caso os valores foram empiricamente definido em um intervalo de (40, 60, 60) a (70, 255, 255) na escala HSV.

2 - A imagem resultante é convertida para tons de cinza.

3 - É aplicado um filtro gaussiano com uma máscara 5x5 com a finalidade de suavizar

possíveis ruídos na imagem.

4 - É aplicada uma limiarização adaptativa na imagem, usando o algoritmo de Otsu. Neste processo os pixels de interesse recebem a cor branca e os demais a cor preta.

5 - É aplicada uma operação morfológica de abertura com o elemento estruturante de tamanho 17×17 , esta operação consiste em uma erosão seguida por uma dilatação na imagem. Neste passo são tirados os ruídos da imagem.

6 - Por fim é realizada uma operação com *AND* lógico na imagem resultante com a imagem original. O resultado final mostra todas as folhas da imagem com um fundo preto.

Na Figura 3.8 é apresentado o resultado da etapa de pré-Processamento.



Figura 3.8: Resultado Pré-Processamento.

- Segmentação: Nessa etapa todas as folhas são divididas em novas imagens, e analisadas separadamente na fase de extração de características. Para chegar a imagem mencionada foi aplicado os seguintes processos:

1 - Foi aplicado a imagem vinda do pré-processamento uma função de dilatação, afim de deixar as bordas da imagem mais grossas.

2 - A imagem foi transformada em tons de cinza.

3 - Foi aplicado uma limiarização na imagem.

4 - Foi usado o filtro de canny (SAHIR, 2019) afim de selecionar todos os contornos dos objetos encontrados na imagem.

5 - Foi aplicada uma função de fechamento, com um elemento estruturante 3x3, afim de melhorar o resultado do filtro de canny, completando possíveis contornos.

6 - Aplicou-se a função findContours da biblioteca OpenCV, afim de armazenar as coordenadas de todos os contornos da imagem.

7 - Foram encontrados todos os pixels dentro de todos os contornos da imagem separadamente, cada contorno gerou uma imagem limiarizada única, dividindo o objeto em questão, do resto da imagem.

8 - Em todas as imagens criadas foi usado uma máscara da imagem original, resultando apenas em um objeto de interesse.

Ao final da etapa de segmentação, uma imagem que antes possuía N objetos, foi transformada em N imagens com apenas um objeto de interesse. Todas as imagens no final do processo são enviadas para a etapa de extração de características.

Na Figura 3.9 é apresentado o resultado da etapa de Segmentação.

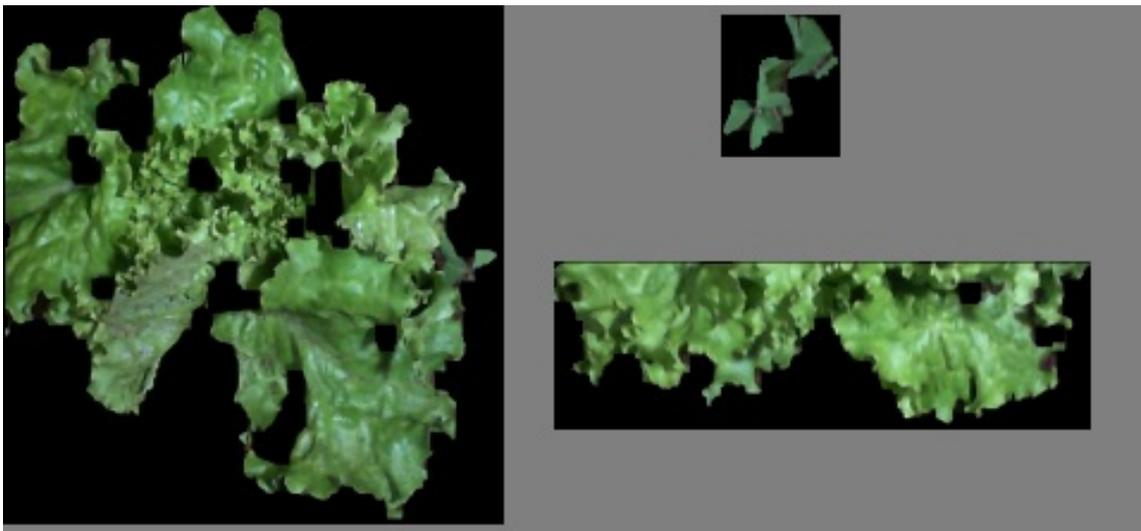


Figura 3.9: Resultado Segmentação.

- Extração de Características: Nesta etapa é feita uma extração das características das ima-

gens, com o método *Local Binary Patterns* (LBP), que corresponde a um descritor local de textura proposto por (OJALA; PIETIKAINEN; MAENPAA, 2002) Sua implementação é baseada na suposição que a informação de uma textura é dividida em padrão e intensidade. Na Figura 3.10 podemos ver, passo a passo, com funciona a análise de um pixel com algoritmo *Local Binary Patterns*. Na abordagem, é escolhido um número fixo de pixels vizinhos calculados, em seguida todos os pixels da imagem são analisados, em cada pixel vizinho ao pixel analisado é atribuído zero caso seu valor seja maior que o pixel selecionado, e um caso o valor seja maior. Esta primeira operação pode ser vista na etapa 1 da figura 3.10. Após a primeira estimativa tais valores são dispostos de forma binária e convertidos para um valor decimal como podemos ver na etapa 2. Por fim, na etapa 3, o valor calculado na etapa anterior é definido com o valor do respectivo pixel analisado, para a matriz de saída.

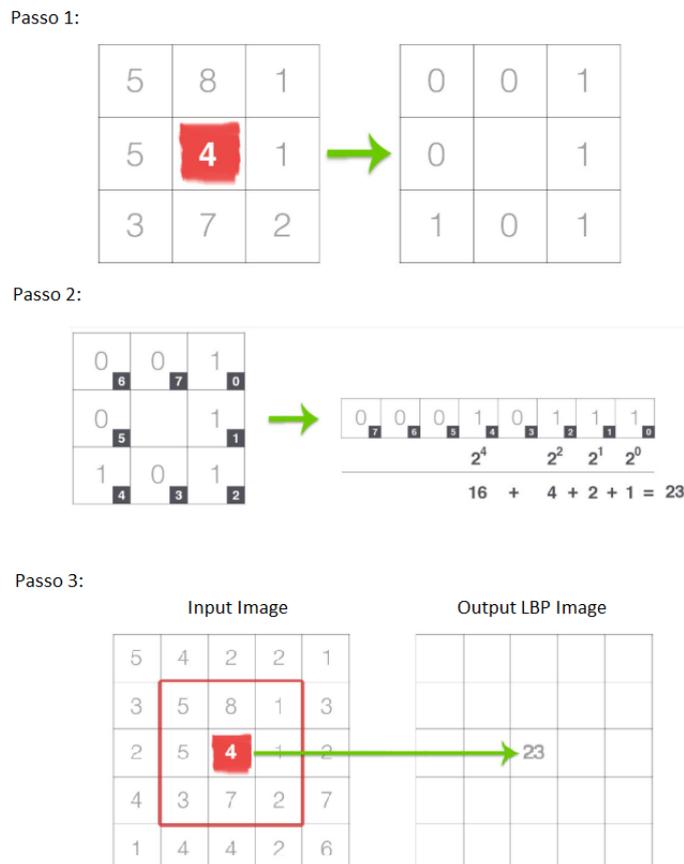


Figura 3.10: Ilustração *Local Binary Patterns*

Para chegar a ao resultado mencionado foi aplicado o seguinte processo:

1 - Foi computado a representação do Padrão Binário Local da imagem.

2 - Em seguida, usou-se a representação LBP para construir o histograma dos padrões.

- **Reconhecimento e Interpretação:** Nesta etapa foi usado uma abordagem com aprendizado de máquina, mais especificamente foi usado uma *support vector machine* (SVM) para reconhecer os padrões e interpretá-los como sendo a cultura desejada ou uma planta invasora. Nesta abordagem o SVM foi implementado usando o valor do parâmetro de regularização de 100 e foi fixado o numero de interações em 10000, isto com os valores vindos do LBP normalizados. No entanto, após resultados negativos foi implementada uma rede neural convolucional afim de garantir melhores resultados. A rede neural foi montada usando duas camadas de convolução, onde as imagens de entrada sofreram uma operação de *resize* ficando com tamanho de 64x64 pixels. Também foram implementadas duas camadas ocultas, cada uma com 128 neurônios e uma camada de saída binaria usando uma função de saída *sigmoid*, para o treinamento foram usadas 7 épocas de treinamento, afim de buscar o melhor resultado. Os passos seguidos para a montagem e treinamento da rede neural convolucional seguem os exemplos de Jones Granatyr (GRANATYR, 2019).

Ao fim desse módulo, o usuário vai receber as informações das porcentagens das plantas invasoras na imagem, e como informação auxiliar o risco que as mesmas representam a cultura analisada, uma vez que dependendo do seu grau de desenvolvimento, plantas invasoras podem representar um risco maior ao cultivo.

3.4 Comunicação do sistema com a internet

A iteração do sistema foi realizada utilizando a biblioteca twitter developers (TWITTER, 2019), que permite postar informações de funcionamento do sistema no formato de um tuíte. Todas as informações são publicadas em tempo real, e são compartilhadas periodicamente durante o dia em formato de status do sistema¹. Um exemplo da abordagem proposta pode ser melhor visualizado na Figura 3.11 . Dentre as informações de status compartilhadas temos: ní-

¹<https://twitter.com/stopasola>

vel de umidade, quantidade de irrigações diárias e a porcentagem de ervas daninhas na cultura e seu respectivo grau de risco ao cultivo. Além das informações do cultivo também é postado uma imagem do mesmo. Além dos tuítes de status do sistema, também é disponibilizado de hora em hora a informação do nível de umidade do solo naquele momento.

Informações do cultivo:
Dia: 20/11/2019 Horário: 20:32
Quantidade de irrigações: 0
Valor umidade solo: 0
Porcentagem de ervas daninhas presentes no cultivo:0.0%

[Translate Tweet](#)



Figura 3.11: Tuíte status do sistema.

3.5 Testes

Esta seção descreve como foi realizado os testes de validação do sistema proposto. É detalhado além da etapa de classificação de imagens o funcionamento da arquitetura implementada.

3.5.1 Sistema de Irrigação

A abordagem utilizada para a irrigação do cultivo se mostrou eficiente, porém ocorreram alguns problemas durante período em que foi testada. O reservatório acima do nível que a cultura foi disposta, permitiu que a água tivesse vazão suficiente para que a mangueira posicionada sobre o cultivo permitisse passagem de água para todas as plantas de maneira satisfatória. Com as informações adquiridas pelo sensor de umidade foi possível o controle

sobre a necessidade de irrigação da planta, entregando o nível de umidade ideal para seu desenvolvimento. Todavia durante a execução do sistema o sensor de umidade utilizado oxidou e parou de funcionar, em seguida foi adquirido um sensor de mesmo modelo para substituir o sensor danificado, no entanto após 7 dias de funcionamento, o novo sensor também oxidou e deixou de funcionar. Após pesquisas constatou-se que devido a certas condições do solo é possível que esse problema ocorra. Verificou-se que existe um modelo de sensor capacitivo, que pode ser protegido por uma película, evitando a oxidação. No entanto, este modelo não foi testado devido ao tempo necessário da aquisição até entrega, inviabilizando os testes. Na Figura 3.12 é possível ver o estado dos dois sensores utilizados no sistema.



Figura 3.12: Sensores danificados.

3.5.2 Sistema de Iluminação

O cultivo foi iluminado através de uma lâmpada de *LED* que emite ondas específicas de crescimento da planta. Dentre as ondas de luz emitidas, temos leds vermelhos para a floração (620-630nm), leds azuis para estimular o período vegetativo (450-460nm), leds infravermelho para estimular brotos e ramificações (730nm), leds UV para atuar na produção de enzimas e afastar pragas e insetos (390nm) e por fim leds branco frio (5000-6000K) entregando o espectro natural da luz, fundamental para o pleno desenvolvimento.

A lampada ficou ligada durante o período de 24 horas por dia, desligando apenas para a aquisição das imagens do cultivo. No entanto após pesquisas constatou-se que para seguir o ciclo de crescimento da plantação, deveria existir um período sem incidência de luz (PAULA, 2015), deste modo optou-se por ligar a lâmpada durante um período de 12 horas diárias.

3.5.3 Sistema de reconhecimento de ervas daninhas

O sistema de reconhecimento funcionou durante um período aproximado de 40 dias, todavia durante esse período houveram momentos em que o protótipo devido a problemas técnicos ficou sem funcionar, impedindo a análise de imagens. Quando em funcionamento as imagens eram adquiridas a cada 4 horas, e exibida no twitter, durante certo período de aquisição das imagens o código responsável pela análise ainda não estava finalizado, deste modo as imagens foram armazenadas no twitter e posteriormente analisadas. Após diversos testes concluiu-se que o descritor de *textura LBP Texture* foi insuficientemente para apontar corretamente a classe (alface ou erva daninha) da imagem analisada, uma vez que segundo o mesmo, todas as imagens pertenciam a uma única classe. Em um primeiro momento atribui-se o problema ao classificador *SVM*, imaginando que o mesmo era incapaz de classificar com precisão o problema analisado. Em um teste adicional, optou-se por utilizar redes neurais convolucionais considerando que seu uso é específico para a análise de imagens. Após a análise do descritor de textura feitos com a rede neural implementada, os resultados obtidos levaram a mesma imprecisão encontrada no *SVM*, portanto dessa forma o descritor utilizado acabou sendo descartado como uma solução viável. Outra solução testada, foi utilizar as imagens como entrada para a mesma rede neural convolucional, no entanto, devido ao pequeno numero de imagens disponíveis o resultado alcançado não foi capaz de descrever com alguma acurácia a rotulação de maneira correta.

Após a eliminação do descritor de textura como solução viável, optou-se pelo uso do serviço em nuvem *Cloud AutoML* (GOOGLE, 2019), que consiste em um sistema de *machine learning* desenvolvido pela Google capaz de realizar a rotulação de elementos em imagens. Para o treinamento do conjunto de dados foi feito o *upload* do banco de imagens já rotuladas no sistema. Por fim a solução final implementada captura a imagem por meio da webcam, em seguida o sistema pré-processa e segmenta a imagem como anteriormente, porém ao invés de realizar a rotulação através do descritor de textura, a imagem é enviada para o preditor executado na nuvem, após o processamento é retornado o rótulo da imagem enviada. Todas as imagens segmentadas da foto original são rotuladas, baseado nas respostas é calculado a porcentagem de ervas daninhas presentes na imagem. Na Tabela 3.1 é possível visualizar os resultados alcançados através do *Cloud AutoML*.

Imagem	Sub Imagens Existentes	Sub Imagens Encontradas	alface	invasora	Fase alface	Porcentagem de Acerto
1	1	1	01--01	00--00	1	100
2	1	1	01--01	00--00	3	100
3	2	2	02--00	00--02	X	0
4	3	2	02--02	00--01	2	66.6
5	3	2	02--02	00--01	2	66.6
6	3	2	02--02	00--01	2	66.6
7	3	2	02--02	00--01	2	66.6
8	3	3	03--02	00--01	2	66.6
9	3	2	02--02	00--01	2	66.6
10	2	2	02--02	00--00	1	100
11	1	1	01--01	00--00	1	100
12	1	1	01--01	00--00	1	100
13	2	2	02--02	00--00	1	100
14	1	1	01--01	00--00	1	100
15	2	2	02--02	00--00	1	100
16	2	2	02--02	00--00	1	100
17	1	1	01--01	00--00	1	100
18	2	2	02--02	00--00	1	100
19	1	5	05--01	00--00	1	0
20	2	2	02--02	00--00	1	100
21	2	2	02--02	00--00	1	100
22	5	5	05--00	00--05	X	0
23	3	3	03--02	00--01	1	66.6
24	3	3	03--02	00--01	3	66.6
25	3	3	03--02	00--01	3	66.6
26	4	3	03--01	00--03	3	25
27	1	1	01--01	00--00	1	100
28	1	1	01--00	00--01	X	0
29	2	1	01--01	00--01	3	50

Tabela 3.1: Resultado rotulação imagens.

Como é possível analisar na segunda e terceira coluna da tabela apresentada, a grande maioria das plantas presentes na imagem foram corretamente segmentadas, resultado este que atingiu uma porcentagem de acerto de 85.48 %. Nos casos em que houve um erro na segmentação o motivo é devido a iluminação da imagem, que devido a uma falha no acionamento da lâmpada de LED, consequentemente atribuiu um espectro de cor diferente do padrão definido, acarretando em uma falha na segmentação do fundo da imagem com as plantas presentes nela. O erro descrito acima pode ser visualizado na Figura 3.13.

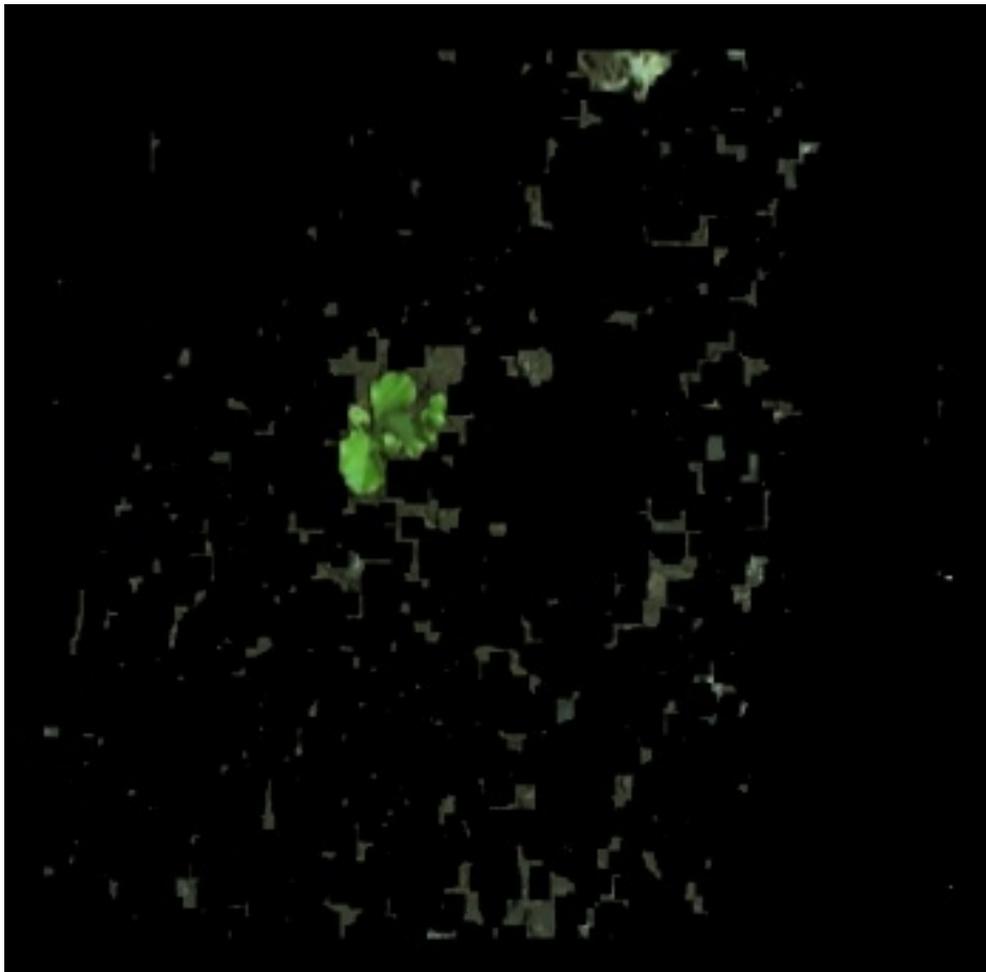


Figura 3.13: Falha segmentação.

Por fim, analisando as colunas 4, 5 da tabela 3.1 é possível analisar os valores a esquerda das colunas, neles está representado a quantidade real de alfaces ou ervas daninhas encontradas na imagem, e o valor a direita (separado por dois traços) representa quantos objetos foram

rotulados, permitindo assim analisar o resultado real X resultado alcançado. Observando a coluna e 7 da tabela concluímos que o *Cloud AutoML* foi impreciso na fase de rotulação das imagens, considerando que todos os objetos destacados na fase de segmentação foram rotulados como alface, porém dentre as 58 sub imagens analisadas apenas 37 de fato eram alfaces, sendo o restante plantas invasoras detectadas de maneira incorreta. Para trabalhos futuros é necessário um aumento substancial do número de imagens, uma vez que no banco de imagens atual existem apenas 131 imagens, divididas em 102 imagens com alfaces e 29 imagens com plantas invasoras, número que se mostrou insuficiente para todas as abordagens testadas. Na coluna 6 da tabela 3.1 é exibido a fase de crescimento da alface presente na imagem, a fase um engloba plantas de 0 a 15 dias, na fase dois temos plantas entre 16 a 30 dias e por fim na fase 3 estão presentes alfaces com 31 a 45 dias. Dentre as imagens analisadas havia fotos apenas com alface, apenas com plantas invasoras e fotos com plantas invasoras e alfaces. O três cenários apresentados podem ser visualizados na Figura 3.14.



Figura 3.14: Combinações entre imagens analisadas.

Capítulo 4

Conclusões

O Sistema proposto atualmente não se mostra uma solução viável para a análise do nível de infestações de plantas invasoras nos cultivos analisados. Os testes realizados contaram com plantas em diferentes diferentes estágios de crescimento porém, para todos os casos, a abordagem que utilizava o detector de textura *LBP Texture* não conseguiu alcançar os resultados esperados, devido a sua incapacidade de distinguir plantas com texturas parecidas. Mesmo após recorrer ao uso da solução *Cloud AutoML* os resultados adquiridos não foram capazes de realizar a rotulação de maneira correta.

Relacionado ao protótipo montado para permitir o crescimento da planta de forma automática, o sistema funcionou de maneira satisfatória, porém apresentou problemas; o mais grave deles foi a dificuldade de crescimento encontrada no ambiente fechado onde o cultivo foi disposto. Após alguns ajustes no tempo de exposição a luz e limiar de umidade utilizado, os resultados apresentados ainda não foram capazes de permitir o desenvolvimento da plantação. Relacionado a arquitetura do protótipo todos os equipamentos funcionaram sem apresentar problemas, com a exceção dos dois sensores de umidade utilizados que sofreram oxidação e deixaram de funcionar, no entanto o problema ocasionado se deu por uma falha do equipamento que deteriorou-se em um curto período de uso.

Em trabalhos futuros uma abordagem com um banco de imagens substancialmente maior, técnicas de extração de características baseadas em tamanho e formato, aliadas a uma rede neural convolucional podem entregar resultados capazes de realizar a descrição correta do tema proposto, uma vez que foi constatado através das abordagens testadas que nesses pontos houveram deficiências. Também nos trabalhos futuros recomenda-se a aquisição de um sensor de umidade resistentes aos efeitos da umidade. Também é recomendado um sensor de fim de curso,

para permitir um melhor controle relacionado a movimentação do motor de passo. Por fim após estas melhorias, futuramente poderá ser implementado uma funcionalidade capaz de eliminar mecanicamente as ervas daninhas presentes no cultivo.

Os códigos utilizados no sistema estão disponíveis em <<https://github.com/Stopasola/Reconhecimento-Vegetais>>, sob a licença Mozilla Public License 2.0 (MPL), a qual garante liberdade de uso e modificação, comercial ou não, desde que se mantenha uma cópia da licença e os devidos créditos aos contribuidores junto ao produto [Mozilla 2012].

Referências Bibliográficas

ANDRIOLO, M. C. G. E. e. M. O. S. J. L. *Crescimento e desenvolvimento de plantas de alface provenientes de mudas com diferentes idades fisiológicas*. 2003. Consultado na INTERNET: <http://www.scielo.br/pdf/cr/v33n1/14140.pdf>.

ARVIND, G.; V.ATHIRA. Automated irrigation with advanced seed germination and pest control. In: *2017 IEEE Technological Innovations in ICT for Agriculture and Rural Development (TIAR)*. [S.l.: s.n.], 2017. p. 64–67.

BANAFSA, A. *What is fog computing?* 2014. Consultado na INTERNET: <https://www.ibm.com/blogs/cloud-computing/2014/08/25/fog-computing/>.

COLLINS, E. The effect of sound on the growth of plants. *Research Gate*, v. 29, 06 2001.

DIAMOND, J. Evolution, consequences and future of plant and animal domestication. *Nature*, p. 700–706, August 2002.

FARMBOT. *Farmbot*. 2014. Consultado na INTERNET: <https://farm.bot/>. Acesso: 12 de abril de 2019.

GOOGLE. *AutoML Vision*. 2019. Consultado na INTERNET: <https://cloud.google.com/automl/>.

GRANATYR, J. *Deep Learning com Python de A a Z - O Curso Completo*. 2019. Consultado na INTERNET: <https://www.udemy.com/course/deep-learning-com-python-az-curso-completo/learn/lecture/10823320content>.

GUTIÉRREZ, J. et al. Automated irrigation system using a wireless sensor network and gprs module. *IEEE Transactions on Instrumentation and Measurement*, v. 63, n. 1, p. 166–176, Jan 2014. ISSN 0018-9456.

KOLMOGOROV; VLADIMIR. Grabcut -interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics (SIGGRAPH)*, August 2004. Disponível em: <https://www.microsoft.com/en-us/research/publication/grabcut-interactive-foreground-extraction-using-iterated-graph-cuts/>.

KWANG; HANG KIM. Support vector machines for texture classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 24, n. 11, p. 1542–1550, Nov 2002. ISSN 0162-8828.

- LAMESKI, P. et al. Cloud-based architecture for automated weed control. In: *IEEE EUROCON 2017 - 17th International Conference on Smart Technologies*. [S.l.: s.n.], 2017. p. 757–762.
- MADALENO, I. M. Cidade das mangueiras. agricultura urbana em belém do pará. *Publisher: Calouste*, August 2002.
- MAZOYER, M. Historia das agriculturas do mundo: do neolítico à crise contemporânea. *Journal: Nature*, August 1997.
- NATIONS, U. *Population Division*. 1999. Consultado na INTERNET: <https://population.un.org/wpp/DataQuery/>. Acesso: 05 de março de 2019.
- NORDIC SEMICONDUTOR. *Specification of the single chip 2.4 GHz Transceiver*. Vestre Rosten 81, N-7075 Tiller, Norway, 2006.
- OJALA, T.; PIETIKAINEN, M.; MAENPAA, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 24, n. 7, p. 971–987, July 2002. ISSN 0162-8828.
- OPENCV. *Biblioteca Opencv*. 2019. Consultado na INTERNET: <https://opencv.org>.
- OVERTON, M. *Agricultural Revolution in England 1500 - 1850*. 2017. Consultado na INTERNET: <http://www.bbc.co.uk/history/british/empireseapower.com>. Acesso: 05 de março de 2019.
- PAULA, R. F. D. *Desenvolvimento*. 2015. Consultado na INTERNET: <http://www.pioneersementes.com.br/silagem/desenvolvimento>.
- PLANTARIO. *Plantário*. 2014. Consultado na INTERNET: <https://www.plantario.com.br/>. Acesso: 12 de abril de 2019.
- POTENA, C.; NARDI, D.; PRETTO, A. Fast and accurate crop and weed identification with summarized train sets for precision agriculture. *Advances in Intelligent Systems and Computing*, 2017.
- PYTHON. *Linguagem Python*. 2019. Consultado na INTERNET: <https://www.python.org>.
- RAO, R. N.; SRIDHAR, B. Iot based smart crop-field monitoring and automation irrigation system. In: *2018 2nd International Conference on Inventive Systems and Control (ICISC)*. [S.l.: s.n.], 2018. p. 478–483.
- SAHIR, S. *Canny Edge Detection*. 2019. Consultado na INTERNET: <https://towardsdatascience.com/canny-edge-detection-step-by-step-in-python-computer-vision-b49c3a2d8123>.
- SANTANDREU, A. Panorama da agricultura urbana e periurbana no brasil e diretrizes políticas para sua promoção: identificação e caracterização de iniciativas de aup em regiões metropolitanas brasileiras. *Revista Geográfica de América Central*, p. 89, Junho 2007.

- SANTOS, A. P. d. O. *Desenvolvimento de descritores de imagens para reconhecimento de padrões de plantas invasoras (folhas largas e folhas estreitas)*. Dissertao (Dissertação de Mestrado) — UFSCAR – Universidade Federal de São Carlos, São Carlos - SP, 2009.
- SARVINI, K. Performance comparison of weed detection algorithms. In: *2019 International Conference on Communication and Signal Processing (ICCSP)*. [S.l.: s.n.], 2019. p. 0843–0847.
- SINGH, P.; SAIKIA, S. Arduino-based smart irrigation using water flow sensor, soil moisture sensor, temperature sensor and esp8266 wifi module. In: *2016 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*. [S.l.: s.n.], 2016. p. 1–4.
- SONKA, V. H. M.; BOYLE, R. *Image Processing, Analysis, and Machine Vision*. 4. ed. [S.l.]: Cengage Learning, 2013.
- SRINIVASAN, N.; PRABHU, P. Design of an autonomous seed planting robot. In: *2016 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*. [S.l.: s.n.], 2016. p. 1–4.
- TWITTER. *Twitter Developers*. 2019. Consultado na INTERNET: <https://developer.twitter.com/en.html>.
- VALA, H.; BAXI, A. A review on otsu image segmentation algorithm. In: *International Journal of Advanced Research in Computer Engineering Technology*. [S.l.: s.n.], 2013.
- WEID, J. M. V. D. Agroecologia: condição para a segurança alimentar. *Agriculturas: experiencias em agroecologia*, p. 4–7, Stembro 2004.
- ZIGBEE. *Zigbee Alliance*. 2019. Consultado na INTERNET: <https://zigbeealliance.org/>.