Universidade Fernando Pessoa

SISAMED: sistema de baixo custo de apoio e alerta à toma de medicação em casa recorrendo a visão computacional e modelos deep learning

Jardson Rodrigues

Faculdade de Ciências e Tecnologia

Orientadores: Prof. José Manuel Torres e Prof. Christophe Soares

Tese submetida ao grau de

Mestre

2022

Resumo

A adesão ao uso de medicamentos é crucial na melhora do quadro de saúde do paciente. Este processo envolve diversos fatores e muitas vezes pacientes interferem diretamente na adesão. Assim é fundamental mecanismos que possam contribuir à terapia. O trabalho apresentado nesta dissertação desenvolve um sistema de reconhecimento automático de objetos no contexto da tomada e adesão ao medicamento. Os objetos detectados são os medicamentos distintos, baseado em análise de sequencias de imagens. A aplicação funciona com base no horário em que cada medicamento precisa ser tomado. Após o cadastro dos devidos dados, o sistema passa a monitorar em tempo real os medicamentos para uma posterior sinalização de que aquele dado horário é necessário uma ação de tomada.

Seguiu-se uma metodologia para o treinamento de um algoritmo para o reconhecimento dos medicamentos utilizando a tecnologia Deep Learning, usou-se as anotações e categorização dos medicamentos desejados para o monitoramento, essas informações foram geradas de uma sequência de imagens e em posições diferentes, afim de gerar um maior dinamismo no reconhecimento. Consideramos que um objeto possa ser deixado sobre a mesa de maneira aleatória. Dados de coordenadas dessas imagens foram carregadas em uma rede neural, utilizando o Framework YoloV4-tiny, que possui poucas camadas convolucionais. Foram utilizados duas classes diferentes de medicamentos para o treinamento e ao final de 4000 interações, 205 imagens e foi alcançado 98.33% de mAP no conjunto de testes dos dois objetos distintos. Duas dimensões foram realizadas no rotulamento das imagens com 150x220 e 400x500. Avaliações realizadas, 86.25% dos casos houve o reconhecimento do medicamento.

Uma abordagem utilizada a tecnologia de Visão Computacional é o treinamento do modelo em um hardware com grande poder de processamento de imagens em GPU. Nesta tarefa de processamento, para alcançar uma maior qualidade técnica, o algoritmo foi submetido em ambiente de nuvem da empresa Google, um espaço disponível e gratuito o suficiente para o treinamento dos objetos em questão. Detalha-se também neste trabalho todas as etapas para o desenvolvimento desta solução e o seu respectivo funcionamento, verificando assim os resultados obtidos por meio de testes perante um cenário

limitado de circunstancias em que o sistema foi submetido. A conclusão mostra que o problema de esquecimento ou não adesão de medicamentos no horário correto possa ser minimizado com tecnologias disponíveis e de fácil acesso, bastando assim um estudo e investigação das mais variadas possibilidades de aplicação de tecnologias de Visão Computacional, que é um dos ramos da Inteligência Artificial. Sua utilização comprova que é possível a classificação de medicamentos utilizando apenas sequências de imagens obtidas por uma câmera de vídeo e submetida a um computador de baixo custo.

Abstract

Adherence to medication use is crucial to improve the patient's health. This process involves several factors and often patients directly interfere with adherence. So are the fundamental mechanisms that can contribute to therapy. The work presented in this dissertation presents an object recognition system in the context of automatic drug taking. The objects detected are the distinct drugs, based on image sequence analysis. The application works based on the time each medication needs to be taken. After registering the appropriate data activation times, the data system moves to a real-time recording.

A methodology was followed to train an algorithm for drug recognition using Deep Learning technology, notes and categorization of the desired drugs were used for monitoring, this information was generated from a sequence of images and in different positions, in order to generate greater dynamism in recognition. We assume that an object can be left on the table at random. Coordinate data from these images were loaded into a neural network using the YoloV4-tiny Framework, which has few convolutional layers. Two different classes of drugs were used for training and at the end of 4000 interactions, 205 images were reached and 98.33% of mAP was reached in the test set of the two different objects. Two dimensions were performed in the labeling of images with 150x220 and 400x500. Evaluations carried out, 86.25% of the cases had the recognition of the drug.

One approach used in Computer Vision technology is the training of the model on hardware with great image processing power on the GPU. In this processing task, to achieve a higher technical quality, the algorithm was submitted to the Google cloud environment, a space available and free enough for the training of the objects in question. This work also details all the steps for the development of this solution and its respective operation, thus verifying the results obtained through tests in a limited scenario of circumstances in which the system was submitted. The conclusion shows that the problem of forgetting or not taking medication at the correct time can be minimized with available and easily accessible technologies, thus, a study and investigation of the most varied possibilities of application of Computer Vision technologies, which is one of the branches of Artificial Intelligence. Its use proves

that it is possible to classify drugs using only sequences of images obtained by a video camera and submitted to a low-cost computer.

Dedico este trabalho primeiramente a Deus.
À minha esposa, e família pelo companheirismo, compreensão e carinho.

Agradecimentos

Primeiramente à Deus, por me conceder saúde e disposição para vencer mais um desafio em minha vida.
Á minha esposa e família, pelo constante apoio, motivação e confiança que me demonstraram em todo momento.
Aos professores Doutores Christophe Soares e José Manuel Torres pela orientação, comprometimento e pelas várias reuniões que foram de suma importância para o desenvolvimento deste trabalho.

Índice

Índice de Figuras

Índice de Tabelas

Lista de Acrónimos

CCD - *Charge Couple Device*

CNN - *Convolutional neural network*

CPU - *Central Processing Unit*

eMAR - *Electronic Medication AdministrationRecord*

GPU - *Graphics Processing Unit*

LCD - *Liquid Crystal Display*

MMU - *Medication Management Unit*

mAP - *Precisão Média*

ONU - *United Nations Organization*

OpenCV - *Open Computer Vison*

RNN - *Recurrent Neural Network*

Raio-X - *X-Radiation*

SISAMED - *Sistema de Alerta de Medicamentos*

TPU - *Tensor Processing Unit*

YOLO - *You Only Look Once*

Capítulo 1

Introdução

A adesão ao tratamento é um fator muito importante para que uma terapia alcance o efeito esperado. Este sucesso inclui, não somente, fazer o uso dos medicamentos nos horários e dias indicados, como também alterar alguns hábitos. A baixa adesão ao tratamento pode provocar a progressão de uma doença, devido o uso incorreto da terapia. Quanto mais adepto o paciente for, melhor para a sua saúde.

Com a evolução contínua da tecnologia, o uso de algoritmos de Inteligência Artificial torna-se cada vez mais uma realidade fundamental nas tarefas diárias. Especificamente no contexto de reconhecimento de medicamentos, ainda não existem soluções suficientemente desenvolvidas que alertem um paciente com doenças crônicas e não crônicas que tomam vários medicamentos que, em um determinado horário um lembrete de tomada de medicamento é acionado com base nos medicamentos sobre uma mesa. Outras soluções foram sugeridas, mas que necessitam de um grande aparato de equipamentos ou não são totalmente autônomos.

Este trabalho propõe uma solução que consiste em desenvolver um sistema baseado em Visão Computacional, utilizando arquiteturas de redes neurais com baixo custo computacional é possível desenvolver software capaz de ajudar pessoas na tomada do medicamento. A aplicação funciona monitorando diretamente e em tempo real os medicamentos com uma câmera de vídeo.

1.1 Motivação e Caracterização do Problema

A razão que motivou a realização deste trabalho é de contribuir para o assistencialismo na saúde e cuidado em casa, desenvolvendo um sistema para ajudar pacientes crônicos ou não crônicos, com um lembrete automático para a tomada do medicamento e identificação de qual medicamento precisa ser tomado. Utilizando uma abordagem com inteligência artificial para a resolução do problema, hardware de baixo custo e treinamento de máquina. O fato de realizar uma pesquisa em tecnologias que podem ajudar na qualidade de vida

torna uma grande satisfação o desenvolvimento deste trabalho.

1.2 Objetivos

Este trabalho foi realizado com o objetivo de garantir a tomada da medicação utilizando um sistema de lembrete. A técnica adotada consiste na aquisição de sequências de imagens da cena em tempo real, detecção e reconhecimento de objetos treinados por um algoritmo de Aprendizagem de Máquina. O sistema consulta uma base de dados local previamente cadastrada, com os medicamentos e horário para efetuar o reconhecimento e disparar o lembrete. A arquitetura do sistema foi desenvolvida para o funcionamento de computadores de baixo custo e uma infraestrutura simples, o algorítimo utilizado pode facilmente ser movido para outros computadores como, Raspberry Pi ou Jetson Nano, como também outros sistemas operacionais.

1.3 Abordagem ao problema e contribuições

O problema abordado neste trabalho consiste em gerenciar os horários e a tomada do medicamento, o problema mais comum na descontinuidade do tratamento é a falta de acompanhamento profissional e ter que usar medicamentos muitas vezes ao dia. Gerenciar os horários e a tomada de vários medicamentos ao dia é um contratempo. Em outros grupos, a baixa adesão estão em pacientes que não têm consciência do esquecimento, desconhecem ou não entenderam as explicações médicas corretamente. O uso de lembretes no celular podem ajudar, porém não podem identificar qual medicamento foi tomado. Portando, um mecanismo que possui a visão e identificação dos medicamentos sobre a mesa, gerenciamento de horário e identificado de qual medicamento precisa ser tomado, pode reduzir consideravelmente os problemas de baixa adesão. Nossa contribuição é a possibilidade de criação de um protótipo que permite o gerenciamento da tomada de medicamentos por meio da Visão Computacional.

1.4 Estrutura do trabalho

Esta dissertação encontra-se organizada em seis capítulos. No primeiro capítulo faz-se uma introdução ao tema desta dissertação, incluindo a caracterização do problema abordado, os objetivos e a motivação para o desenvolvimento do sistema também estão descritas aqui.

No capítulo 2 apresentamos outros trabalhos relacionados ao problema de não adesão ou adesão parcial a medicamentos, como também problemas relacionados ao não cumprimento da tomada correta do medicamento. Descreve-se também às tecnologias utilizadas

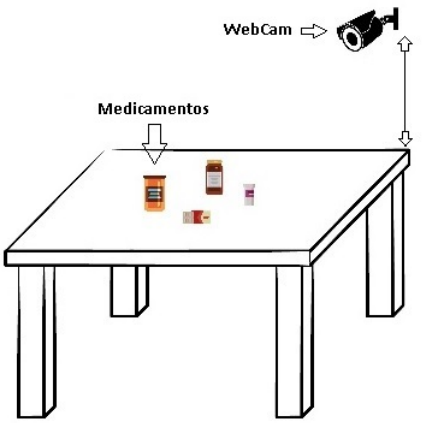


Figura 1.1: Protótipo do Sistema de Reconhecimento

na proposta no desenvolvimento do sistema deste trabalho, teorias de embasamento e métodos utilizados para a criação da solução.

No capítulo 3 é feita uma introdução às tecnologias e especificações de arquitetura do sistema, requisitos e hardware utilizado no desenvolvimento da aplicação. Contexto e abordagem também estão descritas neste capitulo.

O capítulo 4 descreve a implementação do sistema. Apresentam-se a plataforma de desenvolvimento, as bibliotecas e algoritmos utilizados para o funcionamento. A identificação e a técnica utilizada para efetuar o treinamento da máquina para o reconhecimento do objeto. Passos realizados durante a construção e análise de cada processo.

O capítulo 5 descreve os testes e avaliações em que o sistema foi submetido, diversos cenários foram propostos para a simulação de um ambiente real. Testes com o lembrete para o alarme e reconhecimento do medicamento correto para a tomada foi realizado. Tabelas de controle foram informadas para quantificar acertos e erros.

No capítulo 6 apresenta as conclusões relacionadas aos testes em que o sistema foi submetido relatando os experimentos feitos e análise dos resultados obtidos, bem como sugestão para trabalhos futuros e melhorias do sistema.

Capítulo 2

Estado da Arte

2.1 Introdução

Nesta seção apresentamos alguns assuntos e trabalhos relacionados com a tomada de medicamentos. Alguns dispositivos foram propostos na literatura para a melhoria cotidiana do paciente. Em ambientes físicos existem situações em que um equipamento de tamanho considerável, por exemplo uma unidade de gerenciamento de medicamentos, que é conectada diretamente a profissionais da saúde para monitoramento, impossibilita o funcionamento em um ambiente doméstico. Iremos apresentar pesquisas relacionadas a administração de medicamentos e números que comprovam uma melhoria. Trabalhos relacionados ao desenvolvimento de dispensadores de medicamentos serão citados como base de conhecimento.

Com a idade adulta, a adesão aos regimes de medicação prescrita torna-se um dos instrumentos de atividades confusas da vida diária. Os adultos mais velhos com idade a partir de 40 anos geralmente usam mais medicamentos do que os adultos mais jovens com idade entre 20 e 39 anos para tratar doenças crônicas relacionadas à idade, como artrite, doenças cardíacas e diabetes. Pesquisas mostram que os idosos com idade superior a 65 anos tomam em média cinco medicamentos por dia, e esse número pode ser uma estimativa conservadora, dado o foco crescente em agentes farmacológicos para prevenir e tratar doenças crônicas. A má adesão aos medicamentos pode prejudicar sua eficácia em conter a progressão da doença, especialmente se permanecer sem detecção (Carlson et al., 2005).

2.2 Administração de medicamentos

Os enfermeiros precisam fornecer cuidados de alta qualidade, seguros e baseados em evidências. A segurança do paciente e a qualidade do atendimento devem ser prioridades

em todos os momentos. Muitas das tarefas de enfermagem envolvem um certo grau de risco, e a administração de medicamentos apresenta um risco importante. Os enfermeiros tradicionalmente seguem cinco orientações da administração de medicamentos (e.g o paciente certo,o medicamento certo, a dose certa, a via certa e na hora certa) para prevenção de erros (Elliott and Liu, 2010).

Um estudo foi realizado em dois importantes hospitais de ensino em Sydney, Austrália. Foi encontrada uma relação significativa entre interrupções e falhas de procedimento e erros clínicos na administração de medicamentos em ambos os hospitais do estudo. Quanto mais interrupções as enfermeiras recebiam, maior o número de erros. Além disso, descobriram que, à medida que as interrupções aumentavam dentro da administração de um único medicamento, maior a gravidade do erro (Westbrook et al., 2010).

Os resultados de vários estudos feitos em ambientes de internação hospitalar publicados entre 1985 e maio de 2013 foram revisados com o objetivo de avaliar as evidências relacionadas as causas dos erros de administração de medicamentos em ambientes hospitalares. Os deslizes e os lapsos foram os atos inseguros mais comumente relatados, seguidos por erros baseados no conhecimento e violações deliberadas. As condições que provocam erros que influenciam os erros de administração incluem comunicação escrita inadequada (e.g prescrições, documentação, transcrição), problemas com fornecimento e armazenamento de medicamentos (e.g erros de dispensação de farmácia e gerenciamento de estoque em enfermaria), alta percepção de carga de trabalho, problemas com equipamentos baseados em enfermaria (e.g acesso, funcionalidade) , fatores do paciente (e.g disponibilidade, acuidade), estado de saúde da equipe (e.g fadiga, estresse) e interrupções ou distrações durante a administração do medicamento (Keers et al., 2013).

Uma prescrição eletrônica de leitura de código de barras foi utilizada para reduzir erros com dispensação de medicamentos. A intervenção reduziu os erros de prescrição em 47%, de 3,8% para 2,0%. Uma redução absoluta de 1,8% está em linha com a redução de 1,9% (de 6,7% para 4,8%) em um estudo de cuidados intensivos no Reino Unido (Franklin et al., 2007).

2.3 Dispensador de medicamentos

Uma maneira de diminuir os erros na tomada de medicamentos, especificamente para usuários sem supervisão profissional é a utilização de um mecanismo dispensador de medicamentos. Esse tipo de dispositivo pode aliviar as tarefas que são propensas ao erro, realizando a tarefa de administrar a medicação em diversos cenários.

Um sistema com o nome de PillStation fornece uma visualização das pílulas que um paciente deve tomar durante um período de tempo. O dispositivo usa a saída de uma câmera, matriz CCD ou sensor semelhante e dispositivo de captura de imagem, focado no

espaço interior de um compartimento de armazenamento em um dispositivo dispensador, em que a saída é transmitida por linha telefônica comum. O armazenamento é monitorado e a distribuição de medicação inclui vários compartimentos, cada um com um espaço interior para armazenamento de pelo menos um medicamento ou pelo menos um lembrete de medicamento. Um dispositivo de captura de imagem é posicionado para capturar uma imagem do espaço interior de cada compartimento, e um módulo de comunicação transmite a imagem capturada para uma estação de monitoramento central (Bear and Jain, 2011).

Em (Wehba et al., 2014) um projeto de um sistema complexo e bastante abrangente na administração do medicamento foi patentado. A invenção é direcionada em um sistema de gerenciamento que inclui vários computadores, incluindo uma unidade de gerenciamento de medicamentos (MMU) o dispositivo é associado a um médico para realizar um pedido de medicamento. A medicação por sistema de gestão é flexível e integrado para fornecer maior segurança do paciente e produtividade do cuidador com uma variedade de informações hospitalares. O sistema de gestão de medicamentos pode assumir várias formas e fornece métodos para administrar a medicação certa em dose ou taxa certa através do canal ou rota certa do dispositivo (Wehba et al., 2014).

Um dispositivo eletrônico que controla o compartimento de cada medição foi proposto, como se pode observar na figura 2.1 com o uso de uma tela LCD o dispositivo permite que o paciente seja visualmente informado durante a programação de operação e programação do alerta de medicação. A programação utilizada opera com data e etapas de confirmação, permitindo que o paciente adicione informações precisas sobre o monitoramento da medição. Se o paciente não obedecer aos comandos do sistema e não libera o compartimento e o sinal sonoro continuará em períodos pré-determinados (Kehr et al., 1993).

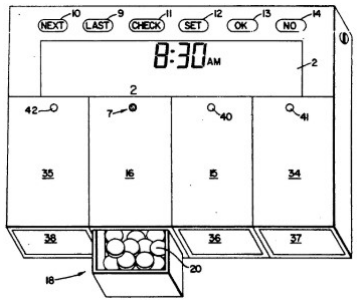


Figura 2.1: Protótipo do Eletronic Medication Dispensing Method Fonte: (Kehr et al., 1993)

Uma tecnologia de código de barras eMAR *(Electronic Medication Administration Record)*, que incorpora várias tecnologias no fluxo de trabalho da equipe de enfermagem para garantir que o medicamento seja administrado no momento correto na dose correta e no paciente correto. Em observações sem a tecnologia utilizada, houve 776 erros de administração de medicamento, uma taxa de 11,5% enquanto que em unidades que uti-

lizam a tecnologia foi observado 495 erros de administração de medicamento, uma taxa de erro de 6,8%, que representa uma redução de 41,4%. Medicamentos com erros de administração fora do horário caiu de 3,1% para 1,6%, representando uma redução de 50,8% (Poon et al., 2010).

Um sistema de gestão de medicamentos foi proposto com uma caixa personalizada para os comprimidos, uma unidade dispensadora de comprimidos, um software para dispensação automática de medicamentos e um modelo de reconhecimento para as pílulas. Uma câmera foi integrada no sistema para adquirir imagens de pílulas, que são transferidas para o aplicativo para reconhecimento da CNN *Convolutional neural network*. Foi desenvolvimento um software aplicativo para uso com um *smartphone* com funções básicas, como cronometragem e contadores. A dispensação de comprimidos requer apenas a definição de sua mistura durante a configuração inicial dos parâmetros. O sistema pode informar o tomador quando tomar o medicamento, e de acordo com os parâmetros definidos, os comprimidos são dispensados separadamente. Devido à necessidade de comparar com imagens de banco de dados ao usar o reconhecimento das pilulas, resultou em um tempo mais longo de processamento (Tsai et al., 2020).

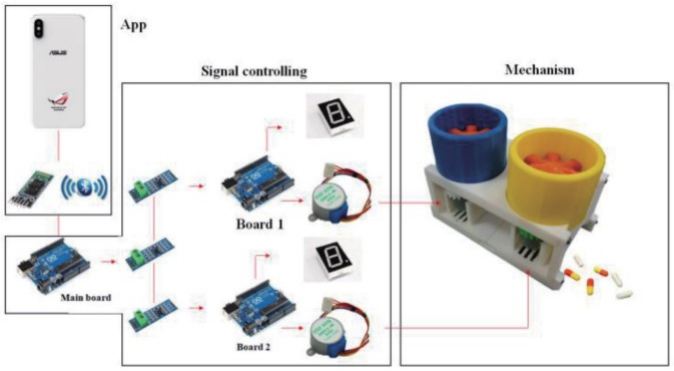


Figura 2.2: Estrutura geral do sistema Smart Pillbox Fonte: (Tsai et al., 2020)

Um sistema de reconhecimento de medicamentos baseado em aprendizagem profunda é proposto, denominado ST-Med-Box. O sistema proposto pretende ajudar pacientes a tomar vários medicamentos corretamente e evitar tomar os medicamentos errados, também pode fornecer outras funcionalidades relacionadas como lembretes para tomar os medicamentos, informações sobre o medicamento e gerenciamento de informações de pacientes. O sistema consiste no reconhecimento dos medicamentos que envia uma notificação ao paciente para tomar sua medicação, o paciente deve colocar o medicamento na região correta do dispositivo proposto e pressionar o botão para iniciar o processo de reconhecimento, após o procedimento o dispositivo irá fornecer uma descrição de voz para explicar o nome, uso, efeitos colaterais e dosagem do medicamento e declarar se o medicamento selecionado é correto. O aplicativo é executado em um dispositivo móvel baseado em Android. O sistema proposto incorpora um sistema baseado em um banco de

dados em nuvem para fornecer aos pacientes serviços de informação adicionais (Chang et al., 2019).

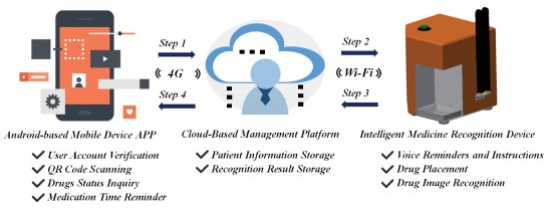


Figura 2.3: Visão geral do sistema proposto ST-Med-Box Fonte: (Chang et al., 2019)

2.4 Não adesão ao tratamento pelos pacientes

A não adesão parcial ou total à terapia anti-hipertensiva está associada a um aumento no risco de evento cardiovascular e morte em todas as idades (Burnier and Egan, 2019). No entanto, o risco é proporcionalmente maior em pacientes idosos, pois eles costumam sofrer de múltiplas comorbidades e têm intrinsecamente um risco cardiovascular maior. Assim, em um estudo de coorte de base populacional de beneficiários de um plano de saúde com idade entre 66-79 anos que foram recentemente diagnosticados com hipertensão e, iniciaram anti-hipertensivos que tiveram incidência de eventos cardiovasculares, foi 2 vezes maior em indivíduos em que o tratamento foi realizado em menos de 80% dos dias (Burnier et al., 2020).

Existe uma grande quantidade de pesquisas relacionadas aos fatores de risco para a não adesão de medicamentos. No entanto, há menos evidências relacionadas a intervenções adequadas para melhorar a adesão e o autocuidado medicamentoso. Além disso, um estudo incluiu uma combinação de intervenções de revisão de medicamentos, modificação de embalagens, educação sobre medicamentos e um gráfico de lembretes de medicamentos (Marek and Antle, 2008).

A falta de adesão terapêutica pode resultar no agravamento do estado de saúde do paciente e da sintomologia da doença, podendo ocasionar igualmente erros no diagnóstico e no tratamento, ou até levar ao seu total fracasso (Cabral and Silva, 2010).

A expectativa de vida ao nascer alcançou 72 anos em 2019, foi adicionado mais 8 anos desde de 1990, conforme a Figura 2.4. Esse número sofre alteração e pode chegar em números maiores em regiões diferentes. É esperado que as melhorias da qualidade de vida continue a subir em todas as regiões, de modo que, em 2050 o a expectativa de vida possa ser projetada em 77 anos globalmente (ONU, 2019).

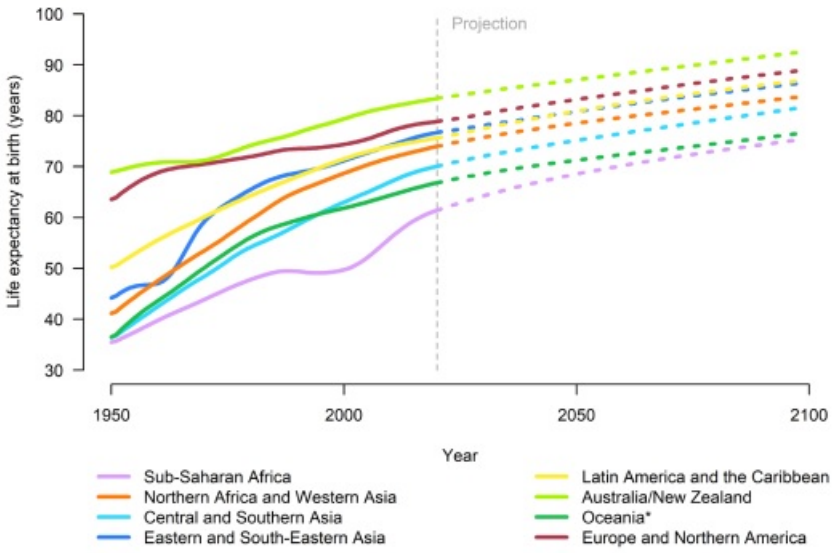


Figura 2.4: Esperança de vida estimada e projetada no nascimento para ambos os sexos pela região SDG. Fonte: ((ONU, 2019))

2.5 A linguagem da tomada de medicamentos

Médicos e funcionários de saúde pública há muito tempo esforçam-se para descrever problemas aos pacientes que se desviam das instruções medicamentosas. Durante os primeiros esforços no controle da tuberculose, dos anos 1890 aos 1940, esses pacientes foram chamados de ignorantes, viciosos, recalcitrantes ou inadimplentes (Roberts, 1953).

A má adesão à medicação tem causas multifatoriais que precisam ser compreendidas antes que as intervenções possam ser manifestadas para melhorar a adesão à medicação (Musoke and Jitta, 1994).

Além de suas implicações sobre o equilíbrio de poder na relação médico-paciente, os termos conformidade e adesão sugerem que as instruções para a tomada de medicamentos são necessárias para atingir os objetivos da terapia (Roberts, 1953).

2.6 Cronofarmacologia

A cronofarmacologia é o estudo de como os efeitos das drogas variam com o tempo biológico e as periodicidades endógenas. O objetivo é melhorar a nossa compreensão das mudanças periódicas e, portanto, previsíveis (por exemplo, circadianas) nos efeitos desejados (cronoefetivos) e na tolerância (cronotolerância) (Reinberg, 1992). O sistema circadiano é responsável por regular uma ampla variedade de ritmos fisiológicos e comportamentais (Ko and Takahashi, 2006).

Uma das características mais importantes do nosso sistema circadiano é que relógios circadianos podem manter o relógio interno abaixo na mudança de luminosidade da es-

curidão constante e sem estímulos externos, sugerindo que nosso corpo tem seus próprios relógios internos (Tahara and Shibata, 2014). Um entendimento completo do ritmo circadiano e sua aplicação para drogas a administração serve à utilização produtiva do medicamento. Deste modo, cronofarmacologia é a sincronização da terapia medicamentosa com o ritmo biológico (Dharani Devangi et al., 2018).

2.7 Processamento de Imagem

2.7.1 Visão computacional

É a capacidade de transformar dados recolhidos por meio de câmera fotográfica ou vídeo permitindo a interpretação e entendimento pelo computador.

A visão computacional possui uma ampla variedade de aplicações como; navegação de robôs, inspeção industrial e inteligência militar e outros como; interação de computadores, recuperação de imagens em bibliotecas digitais, análise de imagens médicas e renderização de cenas sintéticas em computação gráfica (Forsyth and Ponce, 2003).

2.7.2 Imagem digital

Segundo Gonzales e Richard, uma imagem digital pode ser definida como uma função bidimensional,*(x, y)*, onde x e y são coordenadas espaciais e o valor de f em qualquer coordenada de *x, y* é chamada de intensidade ou nível de cinza da imagem no ponto *x, y* e os valores de amplitude de f são quantidades finitas e discretas, chamamos de *imagem digital* (Gonzalez and Richard, 2002).

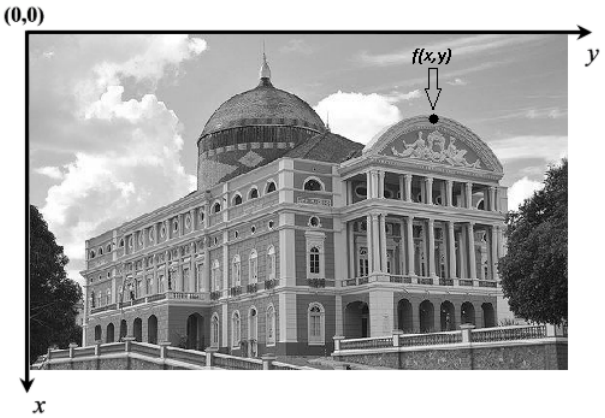


Figura 2.5: Imagem monocromática e a convenção utilizada

A função *f(x,y)* representa o produto da interação entre a iluminância *i(x,y)* que exprime a quantidade de luz que incide sobre o objeto e as propriedades de refletância ou de transmitância próprias do objeto, que podem ser representadas pela função *r(x,y)* cujo

o valor exprime a fração de luz incidente que o objeto vai transmitir ou refletir ao ponto *(x,y)*.(Marques Filho and Neto, 1999)(Szeliski, 2010)

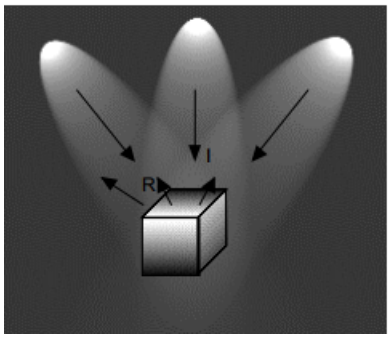


Figura 2.6: Os componentes iluminância (I) e refletância (R) de uma imagem

Para realizar a conversão de uma cena real em uma imagem digitalizada são necessárias duas etapas imprescindíveis, aquisição de imagem e sua digitalização.

2.7.3 Digitalização

O sinal analógico de vídeo obtido à saída do dispositivo de aquisição, deve ser submetido a uma discretização espacial e em amplitude para tomar o formato necessário ao processamento computacional. Do ponto de vista eletrônico, a digitalização consiste em uma conversão analógica digital na qual número de amostras do sinal continuo por unidade de tempo indica a taxa de amostragem e o número de bits do conversor utilizado determina o número de tons de cinza resultantes na imagem digitalizada. Sob uma abordagem matemática formal, o processo de amostragem pode ser visto como uma divisão do plano *XY* em uma grade, com as coordenadas do centro de cada grade sendo uma dupla de elementos do produto cartesiano *ZxZ* (também escrito *Z2*), o qual é o conjunto de todos os pares ordenados dos elementos *a, b* com *a* e *b* sendo números pertencentes a *Z* (conjunto dos inteiros). Portanto *(*x,y) é uma imagem digital se *x,y* forem números inteiros de *ZxZ* e *F* uma função que atribui um valor de nível de cinza (isto é, um número real do conjunto de números reais *R*) para cada par distinto de coordenadas, ou seja, *F* é o processo de quantização descrito anteriormente. Se os níveis de cinza resultantes forem também números inteiros, *Z* substitui *R* e uma imagem digital então se torna uma função bidimensional cujas coordenadas e valores de amplitude são números inteiros (Marques Filho and Neto, 1999)(Sonka et al., 2014).

2.7.4 Biblioteca OpenCV

OpenCV *Open Source Computer Vision Library* é uma biblioteca de visão computacional de código aberto. A biblioteca é escrita em *C* e *C* ++ e funciona em Linux, Windows

e Mac OS X. O Opencv foi projetado para eficiência computacional e com um forte foco em aplicativos em tempo real. Um dos objetivos da biblioteca Opencv é fornecer uma infraestrutura de visão por computador simples que ajude as pessoas a criar aplicativos de visão bastante sofisticados. A biblioteca Opencv contém mais de 500 funções que abrangem muitas áreas da visão computacional, incluindo inspeção de produtos de fábrica, imagens médicas, segurança, interface do usuário, calibração de câmera, visão esteróide e robótica. Opencv tem como objetivo fornecer as ferramentas básicas necessárias para resolver problemas de visão computacional. Em alguns casos, funcionalidades de alto nível na biblioteca serão suficientes para resolver os problemas mais complexos de visão computacional. Mesmo quando este não é o caso, os componentes básicos da biblioteca são completos o suficiente para permitir a criação de uma solução para quase todos os problemas de visão do computador. Opencv também contém uma Biblioteca de aprendizado de máquina (MLL) completa e de uso geral. Esta sub-biblioteca é focada no reconhecimento e agrupamento de padrões estatísticos. O MLL é extremamente útil para as tarefas de visão que estão no centro do funcionamento do OpenCV (Bradski and Kaehler, 2008).

2.8 Aprendizagem de máquina

O aprendizado de máquina é uma área relacionada a Inteligência Artificial, pode ser amplamente definido como métodos computacionais usando a experiência para melhorar o desempenho ou para fazer previsões. A experiência se refere às informações anteriores disponíveis, que normalmente assumem a forma de dados eletrônicos coletados e disponibilizados para análise. Esses dados podem estar na forma de conjuntos de treinamento rotulados por humanos ou outros tipos de informações obtidas por meio da interação com o ambiente (Mohri et al., 2018).

A seguir iremos apresentar conceitos utilizados para o desenvolvimento e funcionamento do sistema proposto. Fundamentos teóricos necessários para o esclarecimento da tecnologia que permite o seu funcionamento.

2.8.1 Treinamento de uma rede

Para que um treinamento realize uma determinada classificação, ele deve ter a decisão desejada da área. Como isso é determinado pelo vetor e limiar de peso, é necessário ajustá-los para obter a funcionalidade necessária. Em termos gerais, ajustando os pesos e limites em uma rede neural geralmente são feitos por meio de um processo interativo de apresentação repetida de exemplos da tarefa requerida. Em cada alteração pequenas alterações são realizadas nos pesos e limites para torna-los mais alinhados com seus valores desejados. Esse processo é conhecido como treinar a rede e o conjunto de exemplos

como o conjunto de treinamento. Do ponto de vista da rede, passa por um processo de aprendizagem ou adaptação ao conjunto de treinamento e prescrição de como alterar os pesos em cada etapa para a regra de aprendizagem (Gurney, 2014).

2.8.2 Aprendizado supervisionado

Consiste em aprender a mapear dados de entrada para alvos conhecidos (também chamados de anotações), dado um conjunto de exemplos que podem ser anotados por *experts*. Geralmente, quase todas as aplicações de aprendizagem profunda que estão em destaque pertencem a esta categoria, como reconhecimento óptico de caractere, reconhecimento de fala, classificação de imagem e tradução de linguagem (Chollet, 2017). Embora a aprendizagem supervisionada consista principalmente em classificação e regressão, existem também outras variantes como:

- Geração de sequência - dada uma imagem, prever uma legenda descrita. Sequência de geração às vezes pode ser reformulada como uma série de problemas de classificação por exemplo; prever repetidamente uma palavra ou token em uma sequência.
- Predição da árvore sintática - dada uma frase, pode prever sua decomposição em uma sintaxe do tipo árvore.
- Segmentação de imagem - dada uma imagem, desenha uma máscara de nível de pixel em um objeto específico.

2.8.3 Rede neural artificial

Uma rede neural é um conjunto interconectado de elementos, unidades ou nós de processamento simples, cuja funcionalidade é vagamente baseada no neurônio animal (Gurney, 2014). A capacidade de processamento de rede é armazenada em intercalação de pontos fortes ou pesos de conexão, obtidos por um processo de adaptação ou aprendizado de um conjunto de treinamento de padrões.

Pode ser definida como uma estrutura de processamento, podendo ser implementada em dispositivos computacionais. O comportamento de uma Rede Neural Artificial vem das interações entre as unidades de processamento da rede.

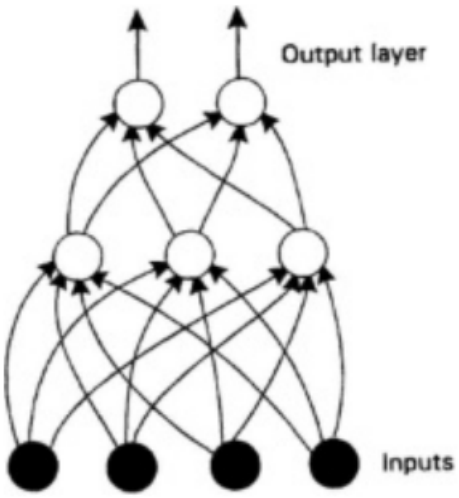


Figura 2.7: Exemplo simples de rede neural
Fonte: (kevin Gurney an introduction to Neural Networks 1997)

Normalmente, os neurônios são categorizados em camadas diferentes, que podem executar transformações diferentes em suas entradas. Os sinais viajam da primeira camada até a ultima camada, possivelmente atravessando as camadas várias vezes. Existem duas categorias principais de arquiteturas de rede, dependendo do tipo de conexões entre os neurônios, ”redes neurais de alimentação para a frente” e ”redes neurais recorrentes”. Se não houver retorno das saídas do neurônio para as entradas em toda a rede, a rede é referida conexão sináptica das saídas para as entradas (suas próprias entradas ou as entradas de outro neurônio), então a rede é chamada de ”rede neural recorrente” (Sazli, 2006).

2.8.4 Redes neurais convolucionais

As redes de convolução combinam três ideias arquitetônicas para garantir algum grau de invariância de mudança, escala e distorção: campos locais receptivos, pesos compartilhados (ou replicação de peso) e subamostragem espacial. Uma rede convolucional típica para o reconhecimento de formas chamada LeNet-5. O plano de entrada recebe imagens de objetos com aproximadamente tamanho normalizado e centralizados (LeCun et al., 1999).

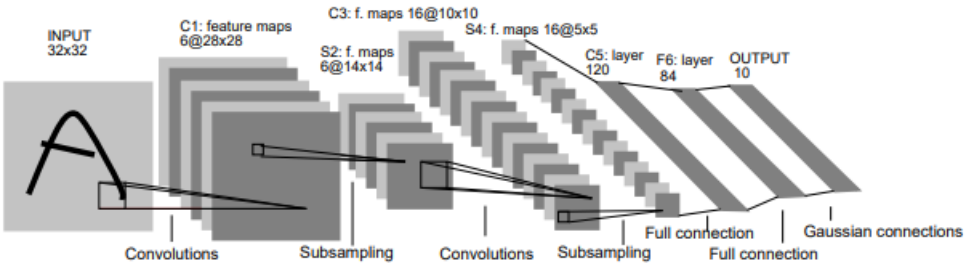


Figura 2.8: Arquitetura do LeNet-5, uma rede neural convolucional, aqui para reconhecimento de dígitos. Cada plano é um mapa, i.e. de características, ou seja, um conjunto de unidades cujos pesos são limitados para serem idênticos.
Fonte: ((LeCun et al., 1999))

O bloco de construção mais importante de uma CNN são os neurônios da camada convolucional, na primeira camada convolucional, não estão conectados a cada pixel na imagem de entrada, mas apenas aos pixels em seus campos receptivos. Por sua vez, cada neurônio na segunda camada convolucional é conectado apenas a neurônios localizados dentro de um pequeno retângulo na primeira camada. Essa arquitetura permite que a rede se concentre em pequenos recursos de baixo nível na primeira camada oculta, em seguida os construa em recursos maiores de nível superior na próxima camada oculta e assim por diante. Esta estrutura hierárquica é comum em imagens do mundo real, que é uma das razões pelas quais as CNNs funcionam tão bem para reconhecimento de imagem (Géron, 2019)(Hassaballah and Awad, 2020).

2.8.5 Framework Darknet

Darknet é uma estrutura de código aberto de alto desempenho para a implementação de redes neurais, cuja arquitetura é chamada de Darknet, que é o mesmo nome do *framework* utilizado para implementar o detector de objetos. Escrito em C e CUDA, pode ser integrado a CPUs e GPUs (Redmon, 2013–2016).

Implementações avançadas de redes neurais profundas podem ser feitas utilizando Darknet. Essas implementações incluem You Only Look Once (YOLO) para detecção de objetos em tempo real, classificação ImageNet, redes neurais recorrentes (RNN).

2.8.6 YOLOv4-tiny

YOLOv4-tiny é proposto com base em YOLOv4 para simplificar a estrutura da rede e reduzir parâmetros, o que o torna adequado para o desenvolvimento em dispositivos móveis e embarcados e melhora a detecção de objetos em tempo real. Primeiramente é utilizado dois módulos ResBlock-D na rede ResNet-D em vez de dois módulos CSPBlock no Yolov4-tiny, o que reduz a complexidade de computação. Em segundo lugar, ele projeta um bloco de rede residual auxiliar para extrair mais informações de recursos do objeto

para reduzir o erro de detecção. No projeto da rede auxiliar, duas convoluções 3x3 consecutivas são usadas para obter campos receptivos 5x5 para extrair recursos globais, canal atenção e atenção espacial também são usadas para extrair informações mais eficazes. Por fim, ele mescla a rede auxiliar e rede de *backbone* para construir toda a estrutura de rede do YOLOv4-tiny (Jiang et al., 2020).

2.8.7 Algoritmo para Treinamento

YOLO é uma abreviatura para o termo '*You Only Look Once*'. Este é um algoritmo que detecta e reconhece vários objetos em uma imagem em tempo real. A detecção de objetos no *YOLOv4* é feita como um problema de regressão e fornece as probabilidades de categorias das imagens detectadas. A arquitetura do *YOLOv4* é composta por *CSPDarknet53* como um *backbone*, módulo adicional de *pooling* de pirâmide espacial, um novo *backbone* que pode aprimorar a capacidade de aprendizado da CNN. O bloco de *pooling* da pirâmide espacial é adicionado ao *CSPDarknet53* para aumentar o campo receptivo e separar os recursos de contexto mais significativos. Em vez de redes pirâmide de recursos para detecção de objetos usados no *YOLOv3*, o *PANet* é usado como o método para agregação e é incorporado principalmente no modelo para aprimorar o processo de segmentação de parâmetros para diferentes níveis de detector.

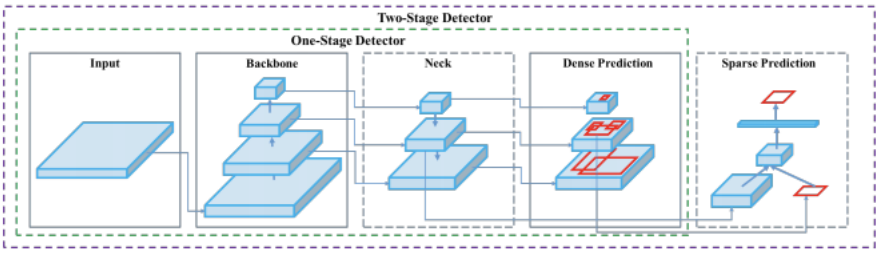


Figura 2.9: Arquitetura do detector Fonte: (Bochkovskiy et al., 2020)

Modelos de estágio único permite que várias previsões no mesmo objeto em uma única imagem, são retiradas as ambiguidades dessas previsões e posteriormente por um processo chamado de supressão não-máxima *NMS*, que apenas deixa a caixa delimitadora com probabilidade mais alta e rótulo para o objeto.

Capítulo 3

Especificação

3.1 Introdução

O sistema proposto neste trabalho sugere um sistema baseado em Inteligência Artificial para ajudar pacientes com doenças a reduzir problemas como a não adesão ou adesão parcial dos medicamentos, a tecnologia proposta utiliza a tecnologia de reconhecimento de imagem baseada em aprendizado profundo para identificar e controlar o gerenciamento da tomada do medicamento no período correto. Este sistema utiliza a aprendizagem da embalagem do medicamento utilizado pelo paciente, que automaticamente identifica o medicamento utilizando uma câmera e controla o horário correto.

3.2 Arquitetura do sistema SISAMED

Esta seção fornece uma visão arquitetural do Sistema SISAMED de forma sucinta, como se pode observar na figura 3.1,os principais elementos que compõe a arquitetura do sistema para a implementação do software. A arquitetura é organizada através de um conjunto de camadas que visam cobrir os principais aspectos técnicos e relevantes ao funcionamento do sistema em questão.

A camada de apresentação exibe uma tela de interação com o cadastro do medicamento e suas informações para o alerta. Também é possível visualizar nessa camada os medicamentos que estão sendo monitorados. Na camada de Modelo Treinado, o arquivo treinado é utilizado para reconhecer determinados padrões nas imagens que são adquiridas por meio da Webcam. Na camada de Negócios não realizadas as validações de horário e existência do medicamento sobre a mesa, essa etapa controla o processo de alerta. Na camada de Persistência as informações de cadastro de reconhecimento dos medicamentos são salvas no Banco de Dados.

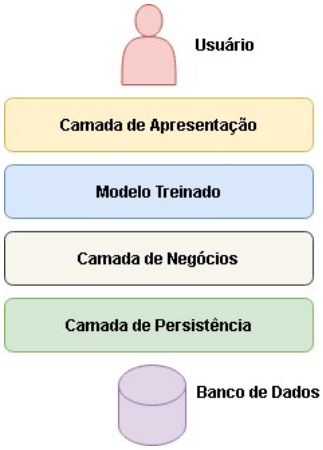


Figura 3.1: Arquitetura do Sistema

Na arquitetura em que o sistema foi desenvolvido permite maior integração e velocidade, pelo fato de que todas as camadas estão no mesmo ambiente desenvolvido, ou seja, no mesmo computador.

3.2.1 Arquitetura do módulo de monitoramento

O módulo proposto oferece uma solução para o monitoramento e controle da tomada de medicamentos por meio da tecnologia de Inteligência Artificial. A ideia central é fazer a detecção e reconhecimento do medicamento e alertar a tomada, a partir deste processo coletar informações para avaliação e controle. Para a detecção de movimento cada medicamento reconhecido é comparado a um *log* e Banco de Dados do sistema. Uma vez que esse medicamento que já estava cadastro não é encontrado na cena, detectamos que houve um movimento com o medicamento.

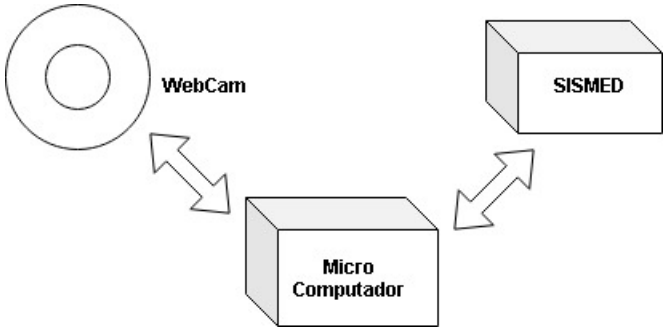


Figura 3.2: Visão da Arquitetura do Sistema

Componentes no Módulo de Monitoramento

- A aquisição da imagem para o monitoramento é feita a partir de um WebCam USB 2.0 Logitech QuickCam Pro 9000 com de foco automático.

- Micro Computador Intel I5 de 4ª geração com 8GB de memória ram, espaço de armazenamento em Hard Disk SSD de 120 GB e Placa de Vídeo NVidea de 2 GB de memória ram.
- Monitor de 18,5" LCD para interação com o sistema de exibição e informações.

3.2.2 Modulo de apresentação

Com base na aquisição da imagem pela Webcam, a mesma é replicada pelo sistema e exibida em um monitor. Essa imagem por sua vez, agora é processada pelo algoritmo e exibe os objetos detectados por meio de um retângulo sobre eles. Na mesma tela informações dos medicamentos são exibidas para controle.

3.2.3 Modulo de negócio

O sistema monitora constantemente o medicamento na cena e compara com o cadastro realizado no Banco de Dados. Para esse funcionamento foi desenvolvido um arquivo de *Log* para salvar essas informações de reconhecimento de cada medicamento no *Frame* da imagem. Dessa forma apenas consultas são realizadas no Banco de Dados *SQLite*. Um algoritmo foi desenvolvido com a finalidade de, se caso a informação já exista no arquivo essa informação não é reescrita, caso contrário, uma nova informação, com o nome do medicamento será escrita e salva no arquivo, para logo em seguida uma consulta de comparação com o Banco de Dados é realizada e assim uma validação é realizada com a hora de alerta cadastrado.

3.2.4 Contexto de utilização do SISAMED

O usuário do sistema possui doenças e toma vários medicamentos ou passa por algum tipo de tratamento em que é necessário a tomada correta do medicamento, este individuo não possui a presença de um profissional da saúde ou mantêm a maior parte do período do dia ou noite sozinho. Na sua casa guarda todos os medicamentos que precisa sobre uma mesa, uma WebCam sobre os medicamentos é posicionada para a aquisição da imagem, um alerta é emitido como lembrete da tomada do medicamento, após remover o medicamento da plataforma é contabilizado um movimento de tomada e assim inativando o lembrete até o próximo momento.

3.3 Requisitos

3.3.1 Levantamento de requisitos

O levantamento dos requisitos do sistema foi realizado baseado em aplicações similares, além da necessidade de controlar a tomada do medicamento de maneira não intrusiva e melhorar a saúde do usuário em questões diante de possíveis falhas ou atrasos no horário estipulado do medicamento. Primeiro foram feitas listas especificando os requisitos funcionais e não funcionais das aplicações que compõem o sistema como um todo. A seguir mostram os requisitos funcionais relativos aos softwares deste projeto.

3.3.2 Requisitos funcionais

[RF1] O sistema é capaz de identificar por meio de uma WebCam o medicamento sobre uma mesa.
[RF2] O sistema de forma automática, alerta o horário correto para a tomada do medicamento.
[RF3] O sistema é capaz de identificar a retirada da caixa do medicamento da cena.
[RF4] O sistema deverá permitir manter um cadastro de medicamentos.
[RF5] O sistema deverá permitir a possibilidade de um histórico de informações sobre interação de cada produto.
[RF7] O sistema deverá permitir que se remova os itens cadastrados, mas sem excluir o cadastro.
[RF8] O sistema deverá permitir que o usuário possa alterar os dados do seu cadastro.

3.3.3 Requisitos não funcionais

[RNF1] O sistema permite o carregamento de novos modelos previamente treinados.
[RNF2] O sistema utiliza código fonte e *framework* gratuito, também permite o funcionamento em hardware de baixo custo.
[RNF3] O sistema deverá operar em arquitetura *desktop*.
[RNF4] O sistema deverá ser desenvolvido na linguagem *Python*.
[RNF5] O sistema deverá utilizar o banco de dados *SQL Lite*.

Conforme os requisitos mencionados e especificados acima, encontra-se agora o modo de interação homem-máquina mais aceitável para uma aplicação do sistema que respeita os requisitos identificados

3.4 Escolha do modelo para o sistema de reconhecimento

Após uma revisão bibliográfica e um estudo em trabalhos relacionados sobre técnicas de monitoramento da tomada do medicamento, foram utilizados dispositivos dos mais variados tamanhos e suportes, incluindo interação remota com o profissional de saúde, o método escolhido proposto para o desenvolvimento e aplicação da detecção do medicamento é o YoloV4-Tiny que propõe em alcançar uma eficiência melhor com recursos limitados. Uma das razões da escolha deste detector e o fato de sua arquitetura de funcionamento permite uma fácil transferência e instalação em outros computadores e sistemas operacionais se assim desejar.

3.5 Conclusão

Neste capítulo foi proposta uma arquitetura do sistema SISAMED que visa o desenvolvimento de uma solução. Regras de negócio e utilização do sistema no contexto foram apresentados. Em seguida iremos descrever a implementação da solução apresentada e todos os passos da sua criação.

Capítulo 4

Implementação

4.1 Introdução

O presente capítulo mostra como foi implementado o algoritmo para o treinamento do modelo. Cada etapa e detalhamento das fases para o desenvolvimento também é informada, visando um melhor entendimento do seu funcionamento. O método baseado em visão computacional e reconhecimento de objetos utiliza uma WebCam para encontrar e classificar o objeto treinado. Esta técnica é precisa, mas requer um maior tempo computacional de processamento devido a utilização para a reconhecimento de objetos simultâneos na mesma imagem. Além disso implementar essa técnica, utilizando redes neurais, requer um custo maior com acessórios de informática como CPU e GPU. O sistema proposto neste trabalho utiliza um equipamento relativamente barato, junto com uma arquitetura de classificação minimalista que consegue alcançar o objetivo de reconhecimento com sucesso. Descrevemos a implementação da aplicação de lembrete para a tomada de medicamentos utilizando a ferramenta Google Colaboratory, para o treinamento deste trabalho é possível utilizar apenas 3 horas de processamento computacional para o treinamento do modelo.

4.2 Treinamento e verificação

Uma das etapas com maior importância é o processo de treinamento utilizando redes neurais artificiais e segmentação dos dados coletados. A captura dos dados para treinamento que servirão para aprendizagem da maquina, ou seja, é associado um conjunto de informações da classe e que cada vetor de descritores representa (rótulos de classes).

Essa etapa consiste em validação é utilizado de amostras para verificar a evolução do treinamento e o desempenho da rede.

4.2.1 Gerando anotações

É relativamente difícil coletar dados de treinamento para segmentação, ou seja, a tarefa de atribuir um rótulo de classe a cada pixel na imagem. Métodos supervisionados requerem um conjunto de imagens de treinamento com anotações por pixel. Para a coleta e criação de um dataset, contendo imagens segmentadas diferenciando por categorias, foi utilizado a ferramenta opensource Labelimg (Tzutalin, 2015), para coletar os dados para o treinamento, a próxima etapa é rotulá-los. No contexto da detecção de objetos, rotular significa desenhar caixas delimitadoras em torno dos objetos que estamos interessados em detectar nas imagens e associá-los às classes de objetos correspondentes para que possamos mostrá-los claramente à máquina. Realizamos as rotulações das imagens em duas dimensões diferentes, 150x220 e 400x500, afim de melhorar o reconhecimento em momentos de redimensionamento.

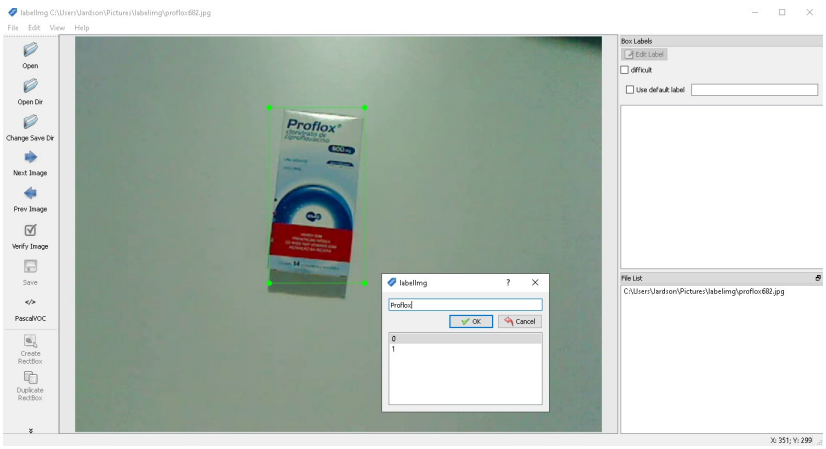


Figura 4.1: Anotação do objeto para treinamento

As posições de anotações também são escritas em intervalos. Ou seja, cada coordenada de caixa delimitadora é dimensionada de zero a um para determinar onde um ponto está localizado na imagem. Por último, como os arquivos de texto que definem as anotações contêm um ID para o objeto que estamos anotando, vemos o ID do objeto declarado primeiro.

4.2.2 Treinamento do modelo

Para o treinamento do modelo, utilizamos uma ferramenta com o nome de Google Colaboratory, é um ambiente de notebook Jupyter on-line baseado em nuvem que nos permite treinar nossos modelos de aprendizagem de máquina e aprendizado profundo em CPUs, GPUs e TPUs. Utilizando a versão gratuita criamos um documento para o desenvolvimento e utilização do algoritmo para o treinamento, esse documento é hospedado no notebook Colaboratory, um ambiente *IDE* de fácil interação que permite escrever e ex-

ecutar códigos em Python. Os notebooks do Colaboratory permitem combinar código executável e comentários, que facilita a documentação e registro de cada etapa do processo de treinamento em um só documento. Na criação de documentos notebooks no Colaboratory, os registros são armazenados na sua conta do Google Drive. Para a facilidade da utilização da ferramenta, o Colaboratory já possuí diversas bibliotecas Python bastando importar ou instalar facilmente, também é possível importar arquivos do Github. Os códigos utilizados nos notebooks do Colaboratory executam códigos diretamente nos servidores em nuvem do Google, sendo possível aproveitar a potência de *Hardware*.

4.2.3 Utilização e configuração Google Colaboratory e YoloV4

Utilizando GPU

A maior vantagem na utilização do Goolge Colaboratory é o fornecimento do suporte gratuito para GPU. Nele é possível selecionar facilmente GPU para o funcionamento do treinamento clicando em Editar, depois Configuração do Notebook em seguida selecionar o acelerador GPU. Não iremos utilizar a TPU nesse projeto, existem outros projetos em que essa funcionalidade é utilizada para a ajudar a otimizar o fluxo de dados.

Figura 4.2: Acelerador GPU

Clonando o Repositório YOLOv4

Para a instalação e utilização do algorítimo para o treinamento é necessário clonar o projeto que está no repositório do Git utilizando no comando:
!git clone https://github.com/AlexeyAB/darknet.

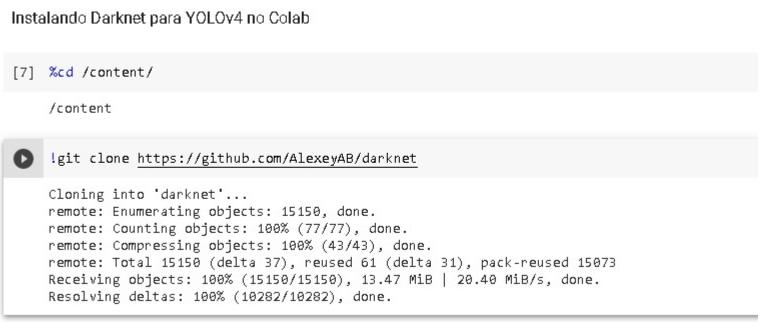


Figura 4.3: Clone git Yolov4

Construindo YOLOv4 usando Make

Mudando para a pasta darknet após o clone, encontramos o Makefile na pasta darknet. É possível ver algumas das variáveis no início do arquivo Makefile. Para construção de GPU defina GPU = 1 e CUDNN = 1 para acelerar na GPU defina CUDNN HALF = 1 para acelerar ainda mais 3 x vezes com precisão mista nos núcleos do tensor. Depois de fazer essas alterações, basta executar o seguinte comando na pasta darknet.

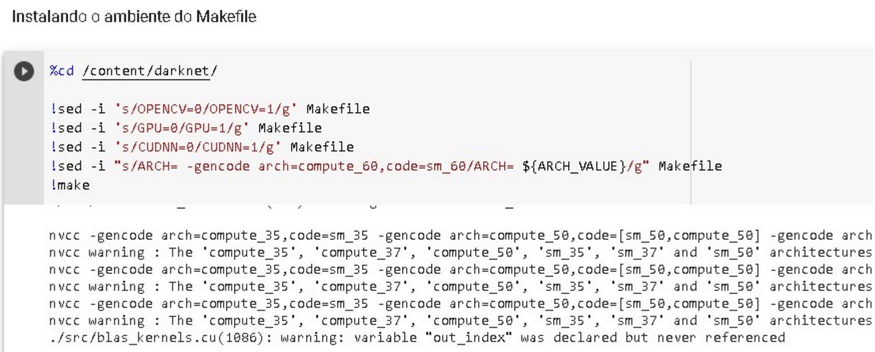


Figura 4.4: Construindo Yolov4

Pesos pré-treinados YOLOv4

Vamos utilizar a aprendizagem por transferência. Em vez de treinar um modelo do zero, utilizamos pesos do YOLOv4-Tiny pré-treinados que foram treinados em até 29 camadas convolucionais. Faremos o download das primeiras 29 camadas do YOLO-Tiny para começar nosso treinamento com os pesos pré-treinados.

Baixando arquivo de pesos YOLOv4

```
%cd /content/darknet
!wget https://github.com/AlexeyAB/darknet/releases/download/darknet_yolo_v4_pre/yolov4-tiny.weights
!wget https://github.com/AlexeyAB/darknet/releases/download/darknet_yolo_v4_pre/yolov4-tiny.conv.29

/content/darknet
--2021-06-27 23:48:26--  https://github.com/AlexeyAB/darknet/releases/download/darknet_yolo_v4_pre/yolov4-tiny.weights
Resolving github.com (github.com)... 192.30.255.112
Connecting to github.com (github.com)|192.30.255.112|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://github-releases.githubusercontent.com/75388965/228a9c00-3ea4-11eb-8e80-28d71569f56c?X-Amz-Algorithm=AW
--2021-06-27 23:48:26--  https://github-releases.githubusercontent.com/75388965/228a9c00-3ea4-11eb-8e80-28d71569f56c?X-A
Resolving github-releases.githubusercontent.com (github-releases.githubusercontent.com)... 185.199.111.154, 185.199.109.
Connecting to github-releases.githubusercontent.com (github-releases.githubusercontent.com)|185.199.111.154|:443... conn
HTTP request sent, awaiting response... 200 OK
Length: 24251276 (23M) [application/octet-stream]
Saving to: 'yolov4-tiny.weights'

yolov4-tiny.weights 100%[===================>]  23.13M  62.0MB/s    in 0.4s
```

Figura 4.5: Pesos Yolov4-tiny

Importar dados no Google Colaboratory

Existem várias maneiras de importar dados com o Google Colaboratory de um Google Drive, incluindo montar seu acesso ao Google Drive na máquina virtual no tempo de execução do notebook Colaboratory, usando PyDrive e usando uma API REST nativa. Neste trabalho vamos utilizar a importação via Google Drive onde nele fizemos o upload dos arquivos necessários para o treinamento.

```
[ ]  from google.colab import drive
     drive.mount('/content/gdrive')
     !ln -s /content/gdrive/My\ Drive/ /mydrive
     !ls /mydrive
```

Figura 4.6: Acesso ao Google Drive

Após a execução do comando, um campo com o link de autorização por meio de uma saída de código e um campo de input é disponibilizado, ao clicar no link é feito um redirecionamento para uma tela seguinte onde é necessário clicar e permitir a liberação por uma conta no Google. Depois disso, aparecerá um código de autorização que você irá copiar e colar na célula.

```
Go to this URL in a browser: https://accounts.google.com/

Enter your authorization code:
```

Figura 4.7: Input de Autorização

Organizando arquivos

Uma vez conectado ao Drive utilizamos o caminho até a pasta onde todos os arquivos de imagem de entrada .jpg e seus correspondentes formatos YOLO foram rotulados como arquivos .txt em uma pasta chamada imgs.

```
[ ]  !unzip /mydrive/yolo-tiny/drugs.zip -d data/imgs
```

Figura 4.8: Descompactação de arquivos

Utilizamos o caminho até a pasta para a criação de um único arquivo com o endereço de todos os arquivos para treinamento, neste trabalho utilizamos cerca de 205 imagens para treinamento, 59 para validação e 29 para teste. O *script* percorre todos os arquivos para a leitura da descrição dos arquivos, e após isso, salva em um arquivo txt para ser utilizado como parâmetro no treinamento e validação.

```
#Configurar diretórios de arquivos de treinamento para conjunto de dados personalizado
%cd /content/darknet/
%cp data/imgs/train/classes.txt data/obj.names
%mkdir data/obj
#copiar imagem e rótulos
%cp data/imgs/train/*.jpg data/obj/
%cp data/imgs/valid/*.jpg data/obj/

%cp data/imgs/train/*.txt data/obj/
%cp data/imgs/valid/*.txt data/obj/

with open('data/obj.data', 'w') as out:
  out.write('classes = 2\n')
  out.write('data/imgs/train = data/train.txt\n')
  out.write('data/imgs/valid = data/valid.txt\n')
  out.write('names = data/obj.names\n')
  out.write('backup = backup/')

#escrever arquivo de treinamento (apenas a lista de imagens)
import os

with open('data/train.txt', 'w') as out:
  for img in [f for f in os.listdir('data/imgs/train') if f.endswith('jpg')]:
    out.write('data/imgs/train/' + img + '\n')

#escreva o arquivo de validação (apenas a lista de imagens)
import os

with open('data/valid.txt', 'w') as out:
  for img in [f for f in os.listdir('data/imgs/valid') if f.endswith('jpg')]:
    out.write('data/imgs/valid/' + img + '\n')
```

Figura 4.9: Organizando arquivos

Detalhes do algorítimo:

- O arquivo obj.names que contém os nomes das classes cada um em uma nova linha.
- Os arquivos train.txt e valid.txt possui o endereço dos aquivos.
- A pasta train para treinamento onde as imagens estão para treinamento e valid que possuí imagens para validação.

4.2.4 Configuração do arquivo CFG

O arquivo .cfg contém todos os parâmetros e configuração do funcionamento da rede neural, nele iremos alterar apenas alguns campos conforme a quantidade de imagens para treinamento. O *script* utilizado automatiza os cálculos para a personalização do arquivo .cfg.

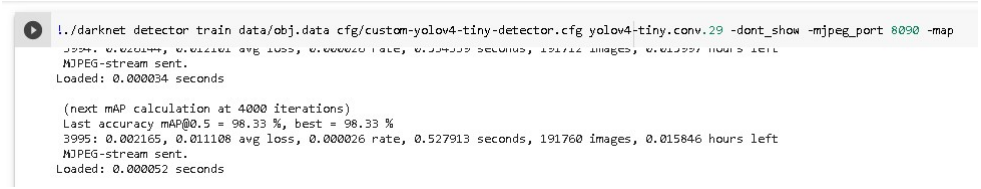
```
#configuração dinamicamente com base no número de classes
def file_len(fname):
  with open(fname) as f:
    for i, l in enumerate(f):
      pass
  return i + 1

num_classes = file_len('data/imgs/train/classes.txt')
max_batches = num_classes*2000
steps1 = .8 * max_batches
steps2 = .9 * max_batches
steps_str = str(steps1)+','+str(steps2)
num_filters = (num_classes + 5) * 3

print("writing config for a custom YOLOv4 detector detecting number of classes: " + str(num_classes))

if os.path.exists('./cfg/custom-yolov4-tiny-detector.cfg'): os.remove('./cfg/custom-yolov4-tiny-detector.cfg')

from IPython.core.magic import register_line_cell_magic

@register_line_cell_magic
def writetemplate(line, cell):
    with open(line, 'w') as f:
        f.write(cell.format(**globals()))
```

Figura 4.10: Script de configuração

Pequenas alterações são realizadas neste arquivo em *max-batches*, *steps-str* e *num-filters*, todos com base no número de classes, para ajustar ao conjunto de dados utilizado.

4.2.5 Iniciando o treinamento

Após a configuração do arquivo CFG e do endereço das imagens, podemos começar a treinar nosso detector de objeto com o comando abaixo.



```
!./darknet detector train data/obj.data cfg/custom-yolov4-tiny-detector.cfg yolov4-tiny.conv.29 -dont_show -mjpeg_port 8090 -map
 MJPEG-stream sent.
Loaded: 0.000034 seconds

 (next mAP calculation at 4000 iterations)
 Last accuracy mAP@0.5 = 98.33 %, best = 98.33 %
 3995: 0.002165, 0.011108 avg loss, 0.000026 rate, 0.527913 seconds, 191760 images, 0.015846 hours left
 MJPEG-stream sent.
Loaded: 0.000052 seconds
```

Figura 4.11: Script de trainamento

A cada 1000 interações é feito um backup do modelo na pasta Backup. O mAP (Precisão Média) será calculado no conjunto de validação e será impresso a cada 4000 iterações.

Precisão da média, é a média de todas as categorias. Tradicionalmente, isso é chamado de, tradução em português de 'precisão média' (mAP). A precisão mede o quão precisas são suas previsões, ou seja, se a porcentagem de suas previsões está correta. O calculo é realizado entre o número de previsões corretas e o número total de previsões. Ao final do treinamento o algorítimo salva na pasta raiz uma imagem da precisão de suas previsões conforme as interações utilizadas. Obtivemos 98.33% de mAP no conjunto de teste e uma perda média de 0.000026.

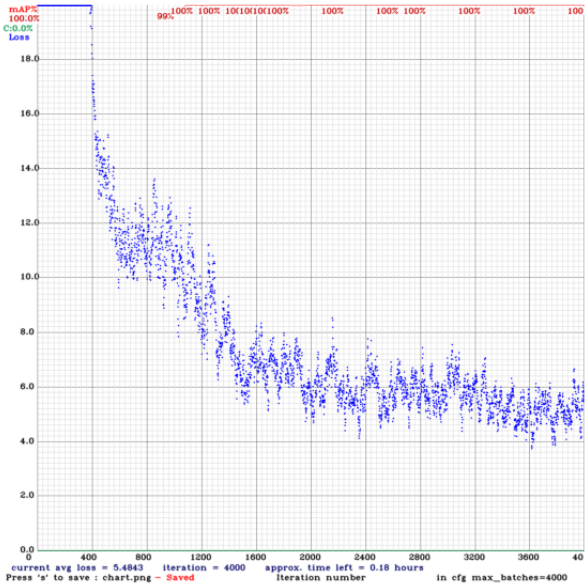


Figura 4.12: Métrica de Precisão

4.2.6 Utilização do modelo

Após um total de 4000 interações o nosso modelo foi treinado para a detecção e reconhecimento do medicamento, essa é a etapa principal para o funcionamento do sistema, visto que é necessário a identificação do medicamento na cena para efetuar o alerta. O treinamento foi realizado com o medicamento parado na imagem, ou seja, apenas identifica e reconhece. Para a detecção do movimento foi desenvolvido um algorítimo que constantemente busca pelos objetos treinados na cena, uma vez que esses objetos não foram identificados, o sistema emite um alerta solicitando a recolocação na cena.

O sistema possui um cadastro dos objetos que precisam ser monitorados, esse cadastro possui um campo de data e hora para efetuar o alerta. Por meio dessa informação o sistema identifica que um determinado medicamento precisa ser removido da cena pelo prazo de 1 minuto, após esse período o sistema solicita novamente que o medicamento precisa ser devolvido a cena e assim identificado que ele foi removido e retornado.

4.2.7 Algoritmo para detecção

Checa arquivo

Foi desenvolvido uma função que verifica se uma determinada *string* existe ou não no arquivo de *log*. A função busca pelo arquivo e aceita uma *string* como argumento. Em seguida busca em cada linha do arquivo se a *string* informada existe ou não. Se no arquivo existir a *string* informada retorna *True* caso contrário retorna *False*.

Desta forma, caso o medicamento não esteja na visão do sistema, essa informação não é escrita no *log* para comparação com o banco de dados.

```
1
2  def check_string(to_search):
3      # Abra o arquivo no modo somente leitura
4      with open('myfile.txt', 'r') as read_obj:
5          # Leia todas as linhas do arquivo uma por uma
6          for line in read_obj:
7              # Para cada linha, verifique se a linha contém a string
8              if to_search in line:
9                  return True
10         return False
```

Figura 4.13: Checa a *string*

Checa visão

No momento da execução do algoritmo, uma função realiza a leitura do medicamento pela Webcam e, conforme o modelo, exibe no nome do medicamento na variável *label*, desta forma, em cada *frame*, o arquivo de log é checado.

```
1      for i in range(len(boxes)):
2          if i in indexes:
3              x, y, w, h = boxes[i]
4              label = str(classes[class_ids[i]])
5              confidence = confidences[i]
6              color = colors[class_ids[i]]
7              cv2.rectangle(frame, (x, y), (x + w, y + h), color, 2)
8              cv2.putText(frame, label + " " + str(round(confidence, 2)),
                   (x, y + 30), font, 2, color, 2)
9
10             if check_string(label):
11                 pass
12             else:
13                 texto = label
14                 arquivo = open('myfile.txt', 'a')
15                 arquivo.write(texto + "\n")
16                 arquivo.close()
```

Figura 4.14: Checa a *label*

Utilizamos um *script* de leitura da imagem da Webcam para o funcionamento e adicionamos a função de *check string* em cada interação.

Inserir no banco

Com base nas informações escritas no arquivo, uma função realiza o *insert* no banco de dados em uma tabela, esse procedimento foi realizado para que quando a comparação for realizada as informações estejam no mesmo banco de dados, porém em tabelas diferentes.

```
def task():
    banco.inserir()
    words1 = (banco.consultar_alerta())
    words2 = (banco.consultar_medicamentos())
    c = set(words1).union(set(words2))
    d = set(words1).intersection(set(words2))
    a = list(c - d)
        if not a:
      print("Ok")
    else:
      print('Falta o {0}'.format(list(c - d)))
```

Figura 4.15: Insere informações no Banco de Dados

Sempre que um *frame* é executado, a tabela no banco de dados é limpa e logo em seguida novas informações são adicionadas. Desta maneira caso um medicamento não esteja na visão esse medicamento não é registrado.

Consulta informações

Após o procedimento de checagem do medicamento na visão da Webcam, essa informação é registrada no banco de dados, dessa maneira comparamos com a tabela de controle, essa tabela informa se o medicamento está faltando na visão e assim é executado um alerta.

```
def task():
    banco.inserir()
    words1 = (banco.consultar_alerta())
    words2 = (banco.consultar_medicamentos())
    c = set(words1).union(set(words2))
    d = set(words1).intersection(set(words2))
    a = list(c - d)
        if not a:
      print("Ok")
    else:
      print('Falta o {0}'.format(list(c - d)))
```

Figura 4.16: Executa Tarefa

Listamos os principais algoritmos para a detecção do medicamento na tela, a solução foi desenvolvida de forma simples, porém bastante eficaz no seu funcionamento.

4.3 Conclusão

Neste capítulo teve por finalidade demonstrar todas as etapas e fases do treinamento do modelo utilizando uma plataforma de nuvem. Na seção de algorítimos para detecção apresenta a forma de contabilização do objeto detectado na cena no sistema local. Este método permite o funcionamento, de modo simples, de toda a regra de negócio do sistema.

Capítulo 5

Testes e Avaliação

5.1 Introdução

Neste capítulo apresenta os testes e avaliações do sistema e o seu funcionamento em condições reais. Está ação é importante para coletar dados estatísticos para o seu aprimoramento. Pretende-se também forçar os limites do seu funcionamento e analisar se de fato o sistema cumpre com o seu objetivo. Os experimentos realizados destinam-se a verificar a funcionalidade e desempenho do sistema proposto, bem como a representação do reconhecimento do objeto que por meio da aprendizagem de máquina, que torna possível o seu funcionamento.

O protótipo com o modelo proposto foi desenvolvido utilizando a linguagem Python. A arquitetura utilizada para o reconhecimento foi o Yolov4-Tiny, devido a sua leveza e capacidade de mobiliada para outras plataformas.

Descrição	Valor
Processador	Intel i5 4590
Memória RAM	8 GB
Disco SSD	120 GB
Sistema Operacional	Ubuntu 18.04.5 LTS
Placa de Vídeo	2 GB

Table 5.1: Especificações da Máquina de Teste

5.2 Sistema de reconhecimento

A Figura 1.1 apresenta o projeto do sistema de reconhecimento desenvolvido, a câmera posicionada sobre os medicamentos possibilita o reconhecimento e detecção de movimento. O movimento é detectado quando o medicamento treinado e cadastrado não é encontrado sobre a mesa. O protótipo em fase de implementação e testes em tempo real

faz com que o sistema realize a leitura da cena e constantemente certifica se o medicamento está presente ou não.

5.2.1 Cenário

O sistema verifica se os parâmetros de horários estão corretos para a tomada do medicamento. A verificação é feita por meio de uma consulta a cada 2 segundos no calendário comparando com o cadastro feito na base de dados. O sistema emite um alerta para a tomada do medicamento conforme todos parâmetros cadastrados, junto com o nome do medicamento cadastrado anteriormente. O alarme continua pelo período de 60 segundos, se dentro desse período o medicamento não for retirado da cena, o alarme irá continuar até o fim do período de 60 segundos, passado assim para o próximo ciclo de monitoramento.

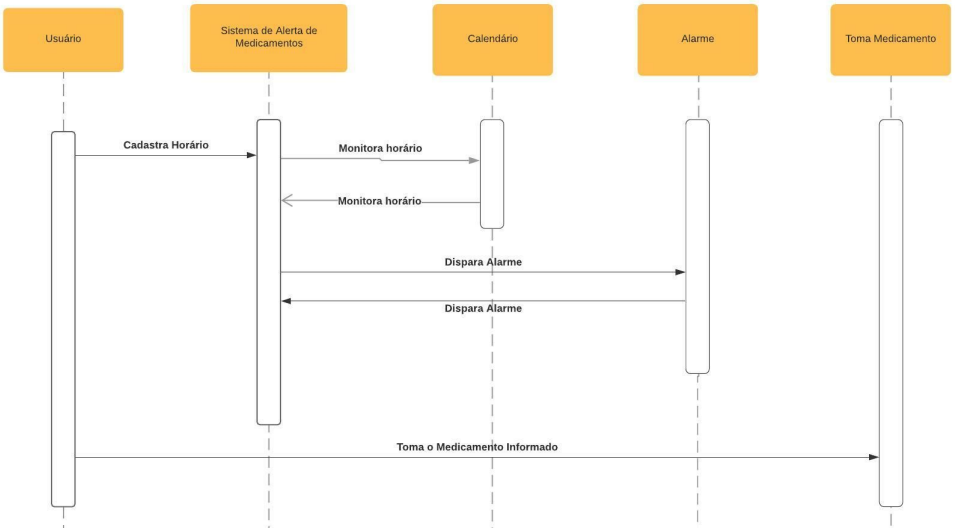


Figura 5.1: Diagrama de sequência para o Cenário

Na figura 5.1 ilustra uma sequência de etapas no cenário sugerido. Usuário realiza uma interação com o sistema, o sistema checa o calendário e posteriormente dispara um alarme. Por fim o usuário toma o medicamento conforme os dados obtidos.

5.2.2 Planejamento da avaliação

O sistema foi desenvolvido de forma simples, com poucos recursos e de instalação fácil. O experimento consistiu na escolha de um local adequado para a instalação, observando a altura em que a câmera precisar ser fixada, de modo a obter um melhor campo de visão e com pouca distância dos objetos de interesse.

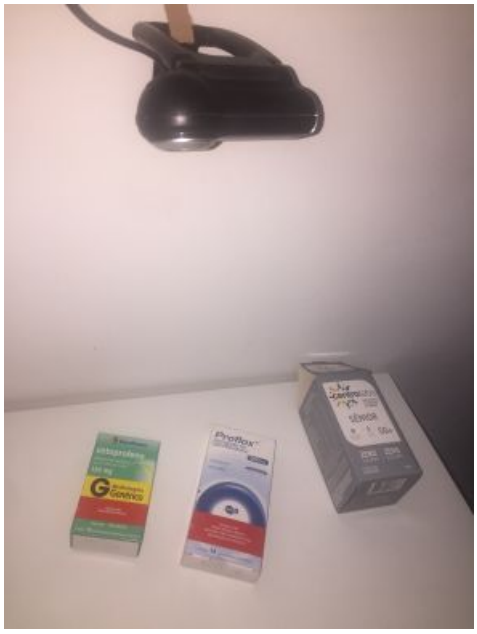


Figura 5.2: Dispositivo desenvolvido

Com uma Webcam sobre a mesa, onde os medicamentos previamente cadastrados estão, com uma altura de até 40 centímetros, que é o suficiente para o funcionamento do sistema. A altura utilizada para a realização dos experimentos foi a mais próxima em que o Dataset para o treinamento foi criado.

Os experimentos realizados nessa seção foram realizados em ambientes internos com presença de luz natural e artificial. O objetivo é reconhecer os medicamentos deixados aleatoriamente sobre uma mesa com outros objetos.

Experimentos com objeto sobre outros, com oclusões, ou parcialmente na cena, impossibilita o funcionamento do sistema. Consideramos que todos os medicamentos estão espalhados de forma aleatória, porém, visível.

5.3 SISAMED: Sistema de Alerta de Medicamentos

O SISAMED *Sistema de Alerta de Medicamentos* é composto por uma tela em que é possível a visualização dos medicamentos reconhecidos pelo sistema. Com o medicamento previamente treinado pelo algoritmo, o sistema identifica e desenha um retângulo no objeto detectado e compara com a base de dados, retornando um status de Ok, uma imagem na cor verde caso tudo esteja de acordo com o funcionamento é exibida na tela.

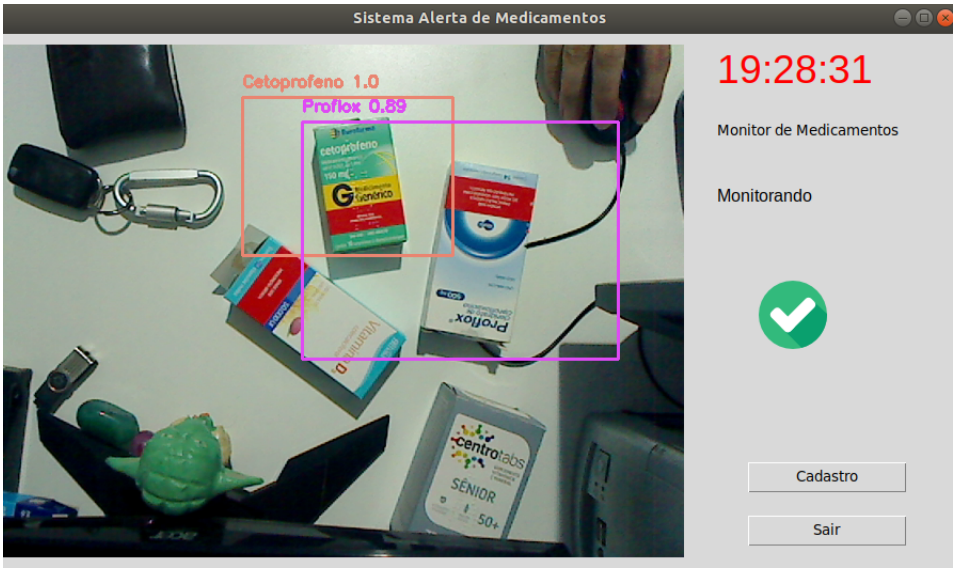


Figura 5.3: Tela Principal do Sistema SISAMED

Na tela principal informações não exibidas para o controle do usuário, também é possível visualizar os medicamentos detectados o horário atual e o *Status* do sistema.

O sistema emite um alerta caso o medicamento não esteja posicionado de forma satisfatória, informando ao paciente que um determinado medicamento está faltando sobre a mesa. O sistema informa qual é o medicamento faltante.

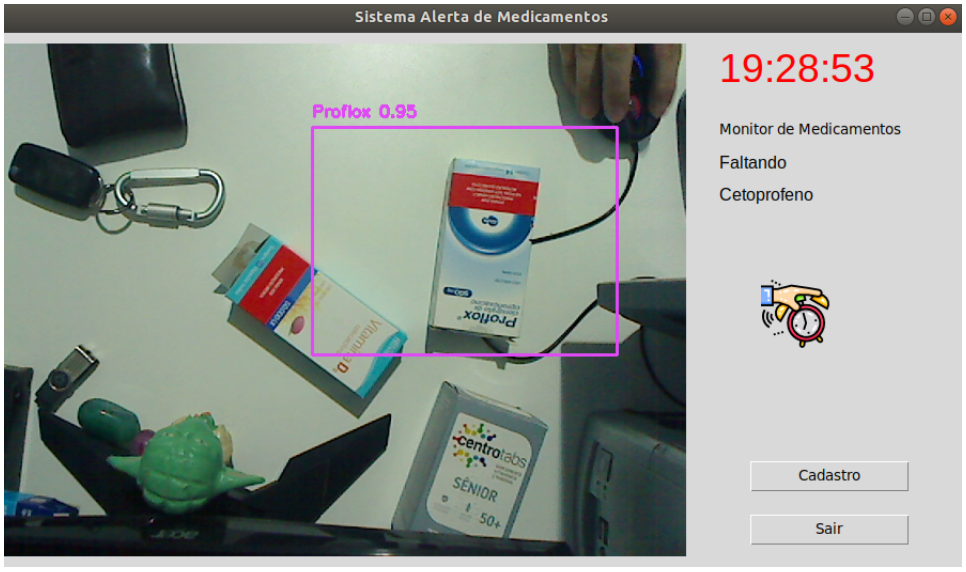


Figura 5.4: Medicamento não encontrado

No lado direito é possível verificar o *Status* do sistema, sinalizando uma mudança de situação.

É possível cadastrar o horário em que um determinado medicamento precisa ser acionado para o lembrete. O sistema monitora os medicamentos reconhecidos, o horário atual e o cadastro de cada medicamento. Quando o horário desejado for igual ao horário atual do

sistema um alerta é disparado com uma imagem de sino informando qual medicamento precisa ser tomado conforme a figura 5.5. O tempo de alerta é de 60 segundos, nesse período o alerta continua com o sino sendo exibido até o momento em que o mesmo for retirado da cena. Após a retirada do medicamento o sistema altera o status para Medicamento Faltando, e informa o nome do medicamento, sugerindo após a tomada do medicamento ele seja reposicionada novamente para o monitoramento.

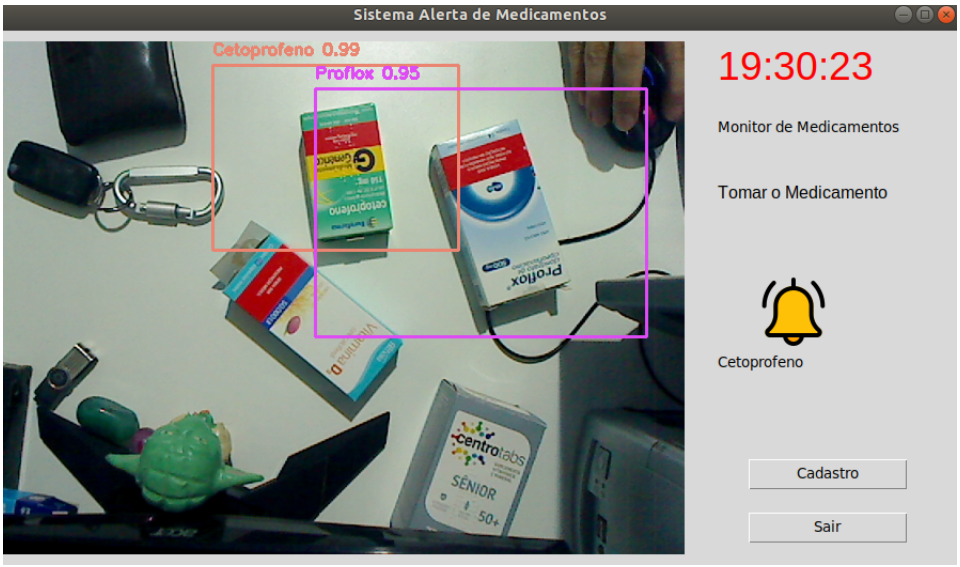


Figura 5.5: Alerta do Medicamento

Após a remoção do medicamento o sistema irá identificar que existe um determinado medicamento faltante e que precisa ser monitorado, isso é necessário para que o monitoramento continue para um próximo ciclo de horário, conforme determinado pelo paciente na tela de cadastro.

Figura 5.6: Medicamento Retirado

Mesmo em casos em que o sistema não encontra na cena os medicamentos treinados para o monitoramento, é acionado um alerta junto com os nomes dos medicamentos. Enquanto o ajuste não for feito o sistema irá continuar com o alerta.

Figura 5.7: Nenhum Medicamento

A seguir serão apresentados os ambientes em que o sistema foi submetido para a simulação. O ambiente para a avaliação, foi criado para uma melhor simulação do mundo real.

5.4 Ambientes de avaliação 1 e 2

5.4.1 Ambiente de avaliação 1

No ambiente para a avaliação 1, a webcam foi posicionada em um altura de 40 centímetros sobre os medicamentos. Os medicamentos estão no centro da imagem adquirida pela Webcam, no momento em que o sistema é carregado os dados de leituras são executados, com isso o funcionamento do sistema é iniciado.

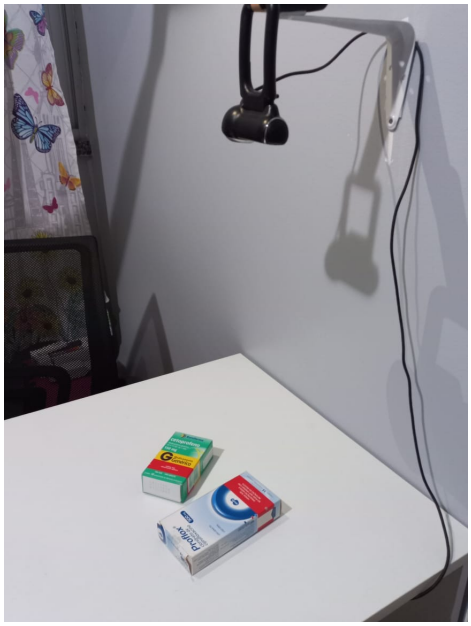


Figura 5.8: Ambiente de Avaliação 1

5.4.2 Ambiente de avaliação 2

No ambiente para a avaliação 2, a WebCam foi posicionada em um altura de 40 centímetros sobre os medicamentos, porém para esta avaliação, a WebCam foi reposicionada em um angulo de 45 graus. Os medicamentos continuam no centro da imagem, porém com uma leve inclinação, no momento em que o sistema é carregado os dados de leituras são executados, com isso o funcionamento do sistema é iniciado.

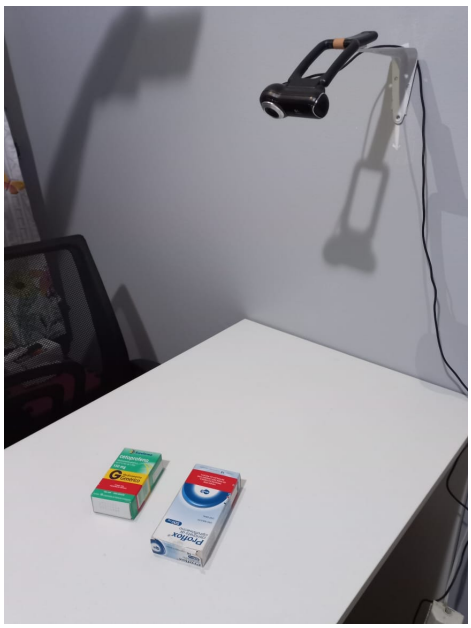


Figura 5.9: Ambiente de avaliação 2

A seguir serão apresentados os resultados referentes aos testes e avaliações. Vídeos foram gravados em um total de 20 minutos, com 8 cenas diferentes. Foram selecionados alguns trechos contendo informações criticas e variadas para demonstrar a eficácia do sistema.

5.5 Avaliações 1 e 2

5.5.1 Avaliação 1

Teste 1

Realizado em ambiente interno, com iluminação artificial, com uma lampada de 9W Led, apresentando apenas os medicamentos treinados sobre a mesa e em um segundo momento outros medicamentos foram adicionados. A cena foi obtida por meio da Webcam instalada em uma altura de 40 centímetros. Os medicamentos estão no centro da cena.

Figura 5.10: Cena 1: Medicamentos treinados sobre a mesa

Controle	Visão Real	Visão SISAMED	Erro no Reconhecimento
Vídeo Cena	Medicamentos 2	Medicamentos 2	0

Table 5.2: Cena 1: Amostragem de Medicamentos treinados sobre a mesa

Teste 2

Outros medicamentos não treinados foram adicionados na cena.

Figura 5.11: Cena 2: Medicamentos treinados sobre a mesa e outros não treinados

Controle	Visão Real	Visão SISAMED	Erro no Reconhecimento
Vídeo Cena	Medicamentos 5	Medicamentos 2	0

Table 5.3: Cena 2: Amostragem de Medicamentos treinados sobre a mesa e outros não treinados

Teste 3

Realizado em ambiente interno com baixa iluminação, com apenas dois medicamentos treinados.

Figura 5.12: Cena 3: Medicamentos treinados sobre a mesa com baixa iluminação

Controle	Visão Real	Visão SISAMED	Erro no Reconhecimento
Vídeo Cena	Medicamentos 2	Medicamentos 2	0

Table 5.4: Cena 3: Amostragem de Medicamentos treinados sobre a mesa com baixa iluminação

Teste 4

Outros medicamentos não treinados foram adicionados na cena.

Figura 5.13: Cena 4: Medicamentos treinados sobre a mesa com baixa iluminação e outros medicamentos não treinados

Controle	Visão Real	Visão SISAMED	Erro no Reconhecimento
Vídeo Cena	Medicamentos 5	Medicamentos 2	0

Table 5.5: Cena 4: Amostragem de Medicamentos treinados sobre a mesa com baixa iluminação e outros medicamentos não treinados

Conclusão

O sistema foi submetido em quatro testes de eficácia no reconhecimento dos medicamentos. Mesmo com situações adversas, em cenário simulado, pôde reconhecer 100% dos medicamentos. O sistema possui uma funcionalidade que caso o medicamento não possa ser reconhecido, mesmo estando sobre a mesa, por motivos de alinhamento, um aviso e emitido de Medicamento Faltando, junto com o nome do medicamento em questão.

5.5.2 Avaliação 2

Teste 1

Realizado em ambiente interno, na mesma situação em que a avaliação 1 foi submetida, com a diferença que, a Webcam foi posiciona em 45 graus sobre a mesa. Os medicamentos treinados foram posicionados em um primeiro momento com poucos objetos, em seguida foi adicionado novos medicamentos para simular um ambiente com uma carga maior de dificuldade.

Figura 5.14: Cena 1: Medicamentos treinados sobre a mesa no ângulo de 45 graus

Controle	Visão Real	Visão SISAMED	Erro no Reconhecimento
Vídeo Cena	Medicamentos 2	Medicamentos 2	0

Table 5.6: Cena 1: Amostragem de Medicamentos treinados sobre a mesa no ângulo de 45 graus

Teste 2

Outros medicamentos não treinados foram adicionados na cena.

Figura 5.15: Cena 2: Medicamentos treinados sobre a mesa no ângulo de 45 graus com outros medicamentos não treinados

Controle	Visão Real	Visão SISAMED	Erro no Reconhecimento
Vídeo Cena	Medicamentos 5	Medicamentos 2	1

Table 5.7: Cena 2: Amostragem de Medicamentos treinados sobre a mesa no ângulo de 45 graus com outros medicamentos não treinados

Teste 3

Realizado em ambiente com baixa iluminação, com a Webcam posicionada em 45 graus, com apenas dois medicamentos.

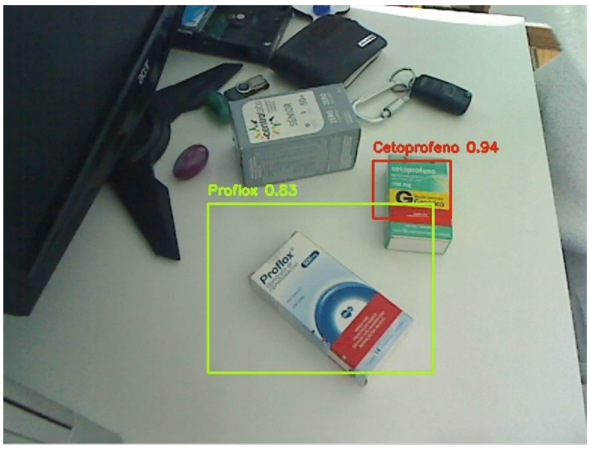


Figura 5.16: Cena 3: Medicamentos treinados sobre a mesa no ângulo de 45 graus com baixa iluminação

Controle	Visão Real	Visão SISAMED	Erro no Reconhecimento
Vídeo Cena	Medicamentos 2	Medicamentos 2	0

Table 5.8: Cena 3: Amostragem de Medicamentos treinados sobre a mesa no ângulo de 45 graus com baixa iluminação

Teste 4

Outros medicamentos não treinados foram adicionados na cena.

Figura 5.17: Cena 4: Medicamentos treinados sobre a mesa no ângulo de 45 graus com baixa iluminação com outros medicamentos não treinados

Controle	Visão Real	Visão SISAMED	Erro no Reconhecimento
Vídeo Cena	Medicamentos 5	Medicamentos 1	1

Table 5.9: Cena 4: Amostragem de Medicamentos treinados sobre a mesa no ângulo de 45 graus com baixa iluminação com outros medicamentos não treinados

Conclusão

O sistema foi submetido em quatro testes de eficácia no reconhecimento dos medicamentos. Com situações adversas, em cenário simulado, houve falha em 13.75% dos medicamentos. Mesmo com o alerta para reposicionar o medicamento, não foi possível obter um resultado 100% satisfatório. Em todos os casos o alarme, conforme o calendário foi disparado. Uma falha foi detectada no não reconhecimento do medicamento, devido a situações e ângulos adversos em que o Dataset foi criado. Em outro momento o sistema não conseguiu diferenciar o medicamento.

5.6 Considerações finais

O sistema desenvolvido para detectar, reconhecer e emitir alerta de lembrete com base em um cadastro, se mostrou bastante eficaz, devido a capacidade computacional utilizada. Quando novos medicamentos foram adicionados na mesma cena, o sistema foi capaz de diferenciar com os medicamentos previamente treinados. Situações como fora do alinhamento, o sistema emitiu alerta para o reposicionamento do medicamento faltante. Em casos em que o posicionamento da Webcam foi em um angulo diferente do frontal, houve variações e erros. O sistema foi submetido em situações desafiadoras em que 86.25% dos casos houve acerto no reconhecimento do medicamento e em 100% o alerta foi emitido, mesmo com o não reconhecimento do medicamento.

Conforme os dados coletados pela análise submetida, definimos que, a melhor maneira de utilização do sistema é o posicionamento frontal da Webcam sobre os medicamentos com uma altura de até 40 centímetros para o pleno funcionamento, dado as circunstâncias e hardware utilizado para o funcionamento do sistema.

Falhas ocorreram devido ao angulo da Webcam de 45 graus e baixa luminosidade, esses elementos não foram coletados para a criação do Dataset para o treinamento do modelo. Esse problema pode ser minimizado utilizando um equipamento mais robusto junto com uma rede neural com mais camadas.

Capítulo 6

Conclusões

Este trabalho apresentou um sistema de reconhecimento de medicamentos integrado a um sistema de alerta que pode ser configurado em qualquer momento pelo usuário. Para isto utilizamos aprendizado supervisionado para o treinamento de objetos com tecnologia de Inteligência Artificial. O sistema apresentado foi desenvolvido em 4 etapas necessárias para o funcionamento. A primeira foi o desenvolvimento de um DataSet com as imagens coletadas dos objetos que serão reconhecidos. Nesta etapa utilizamos um aplicativo para ajudar a criação dos arquivos com as informações de onde cada objeto está na imagem. Nos arquivos são escritos a localização de X e Y de cada objeto em cada imagem. A segunda etapa foi a instalação do *framework* e a estrutura de rede neural onde foi desenvolvido o algoritmo para o treinamento de reconhecido de objetos. A terceira etapa foi o treinamento das categorias de cada objeto desejado, no caso os medicamentos. Foram utilizados nesse trabalho duas categorias diferentes. Na quarta etapa foi utilizado o modelo treinado unindo com um sistema completo onde é possível o gerenciamento de horários para o alerta da tomada do medicamento.

Para avaliar o sistema, foram utilizados 4 cenários diferentes. No primeiro com a câmera focada diretamente sobre os medicamentos, que estavam sobre a mesa, com apenas os medicamentos treinados, gerando uma taxa de reconhecimento de 100% dos casos, junto com o horário de alerta previamente cadastrado. Todos os alertas foram disparados e os medicamentos foram reconhecidos. Na segunda etapa utilizando o mesmo cenário da etapa um, com a diferença que, outros medicamentos não treinados foram submetidos na cena, forçando assim o algoritmo para identificação apenas dos medicamentos treinados. A taxa de 100% do reconhecimento e o disparo do alerta para a tomada do medicamento. O algoritmo foi ajustado para identificar apenas objetos com uma taxa elevada de confiança, mesmo quando o objeto não está alocado corretamente, o sistema dispara um aviso que um medicamento não está sendo reconhecido na cena e informa o nome deste medicamento, solicitando o ajuste caso o mesmo esteja fora do campo de visão da Webcam. Na terceira e quarta etapa, o sistema foi submetido no mesmo cenário da etapa um e dois, com a diferença que, a luz artificial foi parcialmente bloqueada, em 86.24%

os casos houve o reconhecimento do medicamento e o alerta foi disparado. Quando o medicamento não estava de forma satisfatória na cena, o aviso foi disparado solicitando a realocação do medicamento na cena.

Trabalhos futuros

O sistema implementado corresponde a um protótipo e desenvolvimento real de uma proposta para a solução de um lembrete para a tomada de medicamentos baseado em visão computacional. Mesmo após o desenvolvimento e prova de funcionamento há diversas limitações que podem ser aperfeiçoadas para obter um melhor resultado em diversos cenários. O sistema possui limitações em ângulos diferentes e no modo de exibição de alertas, algumas dessas limitações se aplicam na falta de uma GPU considerável. Há também o tamanho do equipamento que pode ser um incomodo no manuseio, podendo ser facilmente substituído por um mini computador Jetson Nano se assim desejar. O sistema foi desenvolvido em uma arquitetura de rede neural leve, por consequência o limita em diferenciar vários medicamentos semelhantes.

Outras possibilidades de desenvolvimento futuro incluem a criação de um dataset com maior número de medicamentos utilizando uma arquitetura de rede neural com mais camadas, associado a um placa de vídeo com GPU. Além disso pretendemos utilizar o sistema em os mais variados cenários e ambientes distintos. Também está previsto uma investigação que possibilite a identificação de padrões de atividades complexas em um longo prazo. O exemplo de utilização seria a contagem de cada medicamento tomado, essas informações poderiam ser relevantes quanto ao fim do tratamento ou lembrete para a aquisição do medicamento para a continuidade do tratamento.

Referências Bibliográficas

Bear, D. M. and Jain, Y. (2011). Medication dispenser with integrated monitoring system. US Patent 8,060,249. 6

Bochkovskiy, A., Wang, C.-Y. and Liao, H.-Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. xi, 16

Bradski, G. and Kaehler, A. (2008). *Learning OpenCV: Computer vision with the OpenCV library*, " O'Reilly Media, Inc.". 12

Burnier, M. and Egan, B. M. (2019). Adherence in Hypertension: A Review of Prevalence, Risk Factors, Impact, and Management, *Circulation Research* . 8

Burnier, M., Polychronopoulou, E. and Wuerzner, G. (2020). Hypertension and Drug Adherence in the Elderly, *Frontiers in Cardiovascular Medicine* **7**(April): 1–9. 8

Cabral, M. V. and Silva, P. A. d. (2010). *A adesão à terapêutica em Portugal: atitudes e comportamentos da população portuguesa perante as prescrições médicas, os hábitos de saúde e o consumo de medicamentos*, ICS. Imprensa de Ciências Sociais. 8

Carlson, M. C., Fried, L. P., Xue, Q. L., Tekwe, C. and Brandt, J. (2005). Validation of the hopkins medication schedule Tc identify difficulties in taking medications, *Journals of Gerontology - Series A Biological Sciences and Medical Sciences* **60**(2): 217–223. 4

Chang, W.-J., Chen, L.-B., Hsu, C.-H., Lin, C.-P. and Yang, T.-C. (2019). A deep learning-based intelligent medicine recognition system for chronic patients, *IEEE Access* **7**: 44441–44458. xi, 8

Chollet, F. (2017). *Deep learning with Python*, Simon and Schuster. 13

Dharani Devangi, R., Shashirekha, C. and Shruthi, S. (2018). A study of chronopharmacological relevance of antihypertensive drugs at a tertiary care hospital-a prospective observational study, *National Journal of Physiology, Pharmacy and Pharmacology* **8**(3): 446–452. 10

Elliott, M. and Liu, Y. (2010). The nine rights of medication administration: an overview, *British Journal of Nursing* **19**(5): 300–305. 5

Forsyth, D. A. and Ponce, J. (2003). *A modern approach*, Prentice-Hall. 10

Franklin, B. D., O'Grady, K., Donyai, P., Jacklin, A. and Barber, N. (2007). The impact of a closed-loop electronic prescribing and administration system on prescribing errors, administration errors and staff time: a before-and-after study, *BMJ Quality & Safety* **16**(4): 279–284. 5

Géron, A. (2019). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*, O'Reilly Media. 15

Gonzalez, R. C. and Richard, E. W. (2002). *Digital Image Processing, Second Edition*, Addison-Wesley. 10

Gurney, K. (2014). *An introduction to neural networks*, CRC press. 13

Hassaballah, M. and Awad, A. I. (2020). *Deep learning in computer vision: principles and applications*, CRC Press. 15

Jiang, Z., Zhao, L., Li, S. and Jia, Y. (2020). Real-time object detection method based on improved yolov4-tiny, *arXiv preprint arXiv:2011.04244* . 16

Keers, R. N., Williams, S. D., Cooke, J. and Ashcroft, D. M. (2013). Causes of medication administration errors in hospitals: a systematic review of quantitative and qualitative evidence, *Drug safety* **36**(11): 1045–1067. 5

Kehr, B. A., Lerner, D., Demenus, R. D. and Edl, M. J. (1993). Electronic medication dispensing method. US Patent 5,200,891. xi, 6

Ko, C. H. and Takahashi, J. S. (2006). Molecular components of the mammalian circadian clock, *Human molecular genetics* **15**(suppl_2): R271–R277. 9

LeCun, Y., Haffner, P., Bottou, L. and Bengio, Y. (1999). Object recognition with gradient-based learning, *Shape, contour and grouping in computer vision*, Springer, pp. 319–345. xi, 14, 15

Marek, K. D. and Antle, L. (2008). Medication Management of the Community-Dwelling Older Adult, *Patient Safety and Quality: An Evidence-Based Handbook for Nurses* pp. 22–30.
URL: *http://www.ncbi.nlm.nih.gov/pubmed/21328769* 8

Marques Filho, O. and Neto, H. V. (1999). *Processamento digital de imagens*, Brasport. 11

Mohri, M., Rostamizadeh, A. and Talwalkar, A. (2018). *Foundations of machine learning*, MIT press. 12

Musoke, R. N. and Jitta, J. N. (1994). Postnatal growth of abandoned preterm babies., *East African medical journal* **71**(8): 519–523. 9

ONU (2019). *World population prospects 2019*, number 141.
URL: *http://www.ncbi.nlm.nih.gov/pubmed/12283219* xi, 8, 9

Poon, E. G., Keohane, C. A., Yoon, C. S., Ditmore, M., Bane, A., Levtzion-Korach, O., Moniz, T., Rothschild, J. M., Kachalia, A. B., Hayes, J. et al. (2010). Effect of bar-code technology on the safety of medication administration, *New England Journal of Medicine* **362**(18): 1698–1707. 7

Redmon, J. (2013–2016). Darknet: Open source neural networks in c, `http://pjreddie.com/darknet/`. 15

Reinberg, A. E. (1992). Concepts in Chronopharmacology, *Annual Review of Pharmacology and Toxicology* . 9

Roberts, F. (1953). Lingua Medica, *Bmj* **1**(4825): 1444–1447. 9

Sazli, M. H. (2006). A brief review of feed-forward neural networks, *Communications Faculty of Sciences University of Ankara Series A2-A3 Physical Sciences and Engineering* **50**(01). 14

Sonka, M., Hlavac, V. and Boyle, R. (2014). *Image processing, analysis, and machine vision*, Cengage Learning. 11

Szeliski, R. (2010). *Computer vision: algorithms and applications*, Springer Science & Business Media. 11

Tahara, Y. and Shibata, S. (2014). Chrono-biology, chrono-pharmacology, and chrono-nutrition, *Journal of pharmacological sciences* **124**(3): 320–335. 10

Tsai, K.-L., Liau, B.-Y., Hung, Y.-M., Yu, G.-J. and Wang, Y.-C. (2020). Development of smart pillbox using 3d printing technology and convolutional neural network image recognition, *Sensors and Materials* **32**(5): 1907–1912. xi, 7

Tzutalin (2015). Labelimg is a graphical image annotation tool and label object bounding boxes in images.
URL: *https://github.com/tzutalin/labelImg* 23

Wehba, S. R., Rinda, J. E., Trohimovich, B. M. and Pelletier, J. (2014). Medication administration and management system and method. US Patent 8,768,719. 6

Westbrook, J. I., Woods, A., Rob, M. I., Dunsmuir, W. T. and Day, R. O. (2010). Association of interruptions with an increased risk and severity of medication administration errors, *Archives of Internal medicine* **170**(8): 683–690. 5