

## Processamento de imagem usando composição de serviços para aplicações de *Smart Farming*

Lucas L. de Jesus<sup>1</sup>, Tenilce G. da S. Alvarez<sup>1</sup>, Júnio C. de Lima<sup>1</sup>, Gabriel da S. Vieira<sup>1</sup>

<sup>1</sup>Núcleo de Informática do Instituto Federal Goiano - Campus Urutaí, GO - Brazil

{lucasdejesus000,tenilcegsalvarez}@gmail.com,

{junio.lima,gabriel.vieira}@ifgoiano.edu.br

**Abstract.** *Service-based systems are used in many application domains, and service compositions designed as the foundation for the full deployment of the Internet of Things and its applications, such as Smart Farming. The main contribution of this work is the study and discussion of the use of image processing services compositions to solve problems of Smart Farming. Initially, two real problems involving image processing are defined and modeled, then these problems are solved using a service composition developed by the authors. The results show the viability of our proposal.*

**Resumo.** *Sistemas baseados em composição de serviços são usados em vários domínios de aplicação e se projetam como fundamento para a implantação plena da Internet das Coisas e suas aplicações, como a Smart Farming ou Fazendas Inteligentes. A principal contribuição desse trabalho é o estudo e a discussão sobre o uso de composições de serviços de processamento de imagem para resolver problemas aplicados na Smart Farming. Inicialmente, são definidos e modelados dois problemas reais que envolvem processamento de imagem, em seguida, esse problemas são resolvidos usando uma composição de serviços elaborada pelos autores. Os resultados mostram a viabilidade da nossa proposta.*

### 1. Introdução

A Computação Orientada a Serviços (*Service-Oriented Computing* - SOC) e sua realização através de tecnologias de serviços *web* oferecem uma solução robusta para a integração de aplicações com o objetivo de criar novos serviços com valor agregado. Serviços *web* têm sido amplamente utilizados na realização de diversas tarefas e, cada vez mais, são vistos como uma boa prática de promover a reutilização de recursos e integração entre sistemas. Nesta nova realidade, a funcionalidade que um serviço *web* isolado oferece nem sempre satisfaz as necessidades das aplicações ou reflete de maneira apropriada complexos processos de negócios da atualidade [Alonso et al. 2004].

Serviços *web*, ou simplesmente *serviços*, nesse contexto, podem ser definidos como entidades de *software* autocontidas que executam uma ou mais operações, cujas interfaces são publicadas, e que oferecem suporte para interações diretas com outras aplicações [W3C 2002]. De acordo com SOC, o foco do desenvolvimento muda de atividades relacionadas ao projeto e a implementação de componentes de sistemas personalizados para atividades relacionadas à identificação, seleção e composição dos serviços oferecidos por terceiros [Cardellini et al. 2012]. Uma composição de serviços é uma

combinação de múltiplos serviços, que podem ser oferecidos por provedores diferentes e em locais diferentes na rede, para executar um conjunto de atividades que satisfaçam um objetivo específico [Baryannis et al. 2010].

Aplicações construídas como composições de serviços são especificadas como processos abstratos compostos, os quais, por sua vez, são constituídos por um conjunto de atividades que devem ser desempenhadas pelos serviços, bem como o inter-relacionamento entre as atividades. Em seguida, cada atividade definida em uma composição deve ser desempenhada por um serviço concreto previamente selecionado, assegurando os requisitos dos usuários [Baryannis et al. 2010].

Vale ressaltar que uma atividade pode ser desempenhada por diversos serviços, desde que estes implementem os requisitos funcionais esperados. Quando um usuário envia uma mensagem para uma composição, os serviços selecionados iniciam o processo de troca de mensagens a fim de desempenhar a lógica de negócio descrita pela composição. Uma composição de serviços é tipicamente opaca para o usuário, que a considera como um serviço único e não um conjunto de serviços distintos [Furtado et al. 2014].

O poder de compor serviços independentes em serviços com maior granularidade promove mais produtividade e reutilização, onde os serviços podem ser utilizados para agasalhar aplicações existentes, bem como para desenvolver novas aplicações. Nesse sentido, recentes avanços no desenvolvimento de sistemas de *software* evidenciaram uma mudança de paradigma, partindo de soluções convencionais em direção à adoção de aplicações baseadas em serviços, criadas usando implementações sob medida ou utilizando componentes previamente desenvolvidos. Portanto, composições de serviços podem ser empregadas nas mais diversas áreas como serviços de *streaming*, serviços de nuvem e Internet das Coisas.

Segundo Bertolino *et al.* [Bertolino et al. 2011], o desenvolvimento de sistemas na era Internet das Coisas deve ser capaz de abordar a complexidade, a heterogeneidade, a interdependência e, especialmente, a evolução de sistemas em rede fracamente conectados. Sendo assim, aplicações baseadas em composições de serviços se projetam como fundamento para consolidação da Internet das Coisas [Issarny et al. 2011, Autili et al. 2014]. Dessa forma, a Internet das Coisas é também algo essencial para uma agricultura inteligente, também chamada de *Smart Farming*, tornando possível o aumento da produtividade e diminuição dos custos guiados por ferramentas computacionais que garantem de fato controle e melhores resultados.

A área de *Smart Farming* é habilitada por sistemas agrícolas automatizados, constituídos com diversos dispositivos e atuadores de sensores sem fio, capazes de monitorar as condições ambientais e controlar os dispositivos implantados de acordo com os dados coletados por meio de redes de acesso com e sem fio [Ryu et al. 2015]. Dentre as aplicações para *Smart Farming* destacam-se aplicações que utilizam técnicas de processamento de imagem para tomada de decisões. Geralmente as imagens podem ser capturadas por satélites, drones e, inclusive, por pessoas com câmeras especiais ou convencionais. As imagens capturadas podem ser processadas e, junto com demais componentes, podem ser utilizadas para descoberta de queimadas, monitoramento no crescimento das plantas e detecção de doenças em frutas, por exemplo. Sendo assim, o uso de composição de serviços nesse contexto pode reduzir o custo e os riscos da construção de novas aplicações, uma vez que as lógicas de negócios existentes são representadas como serviços e podem

ser reutilizadas.

Entretanto, construir e implementar sistemas que necessitam de processamento de imagens com a adoção de conceitos de composição de serviços (como reutilização, compartimentação e escopo extensível) não são amplamente discutidos na literatura. Artigos e relatórios técnicos enfatizam o problema de processamento de imagem, mas, frequentemente, os principais componentes de *software* para a construção de tais aplicações são distribuídos sem muito destaque sobre seus detalhes técnicos, inclusive sem apresentar a solução em uma linguagem de programação de alto nível. Por exemplo, para problemas que envolvam processamento de imagem é utilizado o Matlab, um ambiente mais adequado para prototipação de soluções. Esse problema é agravado ainda mais quando diferentes abordagens precisam ser implementadas juntas em um mesmo aplicativo para compartilhar recursos e código.

Assim, para contribuir com a solução desse problema, no presente trabalho, apresentamos um estudo que inclui uma delimitação e uma discussão sobre o escopo de processamento de imagem, na qual serviços podem ser disponibilizados por provedores e acessados por terceiros de forma a permitir a construção de aplicações baseadas em serviços. Nesse sentido, nosso estudo mostra dois experimentos que lidam com problemas que necessitam de processamento de imagem, resolvidos por meio de composição de serviços para *Smart Farming*.

O restante desse trabalho é organizado como se segue. Na Seção 2 é realizado uma revisão dos principais trabalhos que lidam com processamento de imagem para área de *Smart Farming*. Em seguida, na Seção 3, os resultados do delineamento metodológico são apresentados juntamente com a definição dos principais conceitos empregados. Na Seção 4 são apresentados os resultados dos experimentos realizados em uma composição de serviços para *Smart Farming* elaborada pelos autores. Por fim, na Seção 5 são discutidas as conclusões e os trabalhos futuros são apresentados.

## 2. Trabalhos relacionados

Na *Smart Farming* existem vários modelos computacionais que podem ser utilizados em conjunto para o desenvolvimento de softwares que auxiliem na gestão da agropecuária [Honda and JORGE 2013]. Inclusive técnicas de processamento de imagens em conjunto com sensores e sistemas de posicionamento global podem ser utilizados para auxiliar na agricultura de precisão. Entretanto, as aplicações nessas duas áreas quase sempre são construídas com um alto acoplamento entre as funcionalidades, o que dificulta a sua utilização levando em conta o reuso e a distribuição dos serviços, premissas da Internet das Coisas.

Exemplos disso são resultados produzidos por Yang *et al.* [Yang et al. 2007], que exploraram técnicas de processamento de imagem e câmeras estéreo para detectar tomates, e Çakir *et al.* [Çakir et al. 2013], que utilizam técnicas de processamento de imagem para detecção de laranjas. Já o trabalho de Jhuria *et al.* [Jhuria et al. 2013] utiliza processamento de imagens como uma ferramenta para monitorar as doenças em frutas durante o cultivo, desde a plantação até a colheita.

O trabalho de Yang *et al.* [Yang et al. 2000] apresenta uma metodologia para o processamento de imagens obtidas a partir de milharais a fim de determinar um mapa de plantas daninhas e simular o controle de um pulverizador de herbicidas. O trabalho

de Fan *et al.* [Fan et al. 2017] utiliza imagens hiper-espectrais combinadas com quimiometria para detectar infestações precoces de brocas em plantações de arroz, bem como identificar o grau de infestação como forma de garantir a proteção do crescimento do arroz. Todos esses trabalhos discutidos aplicam técnicas de processamento de imagem sem se preocupar com a integração com outras aplicações e/ou recursos de terceiros.

Recentemente alguns trabalhos buscam desenvolver soluções para processamento de imagem através de técnicas de integração de aplicações. Por exemplo, Rupanagudi *et al.* [Rupanagudi et al. 2015] desenvolveu uma aplicação para detecção de pragas em tomates através do monitoramento constante usando processamento de vídeo com o uso de recursos fornecidos por computação em nuvem. Já Vieira *et al.* [Vieira et al. 2019] propôs uma arquitetura de software, denominada *Disparity Computation Framework* (DCF), que pode acomodar diferentes métodos para lidar com a construção de mapas de disparidade dado duas imagens. As principais contribuições deste estudo incluem uma delimitação e uma discussão sobre o escopo do mapa de disparidade, na qual componentes estruturados podem ser organizados em camadas, sendo capaz de permitir o reaproveitamento entre diferentes aplicações.

Entretanto, trabalhos direcionados a resolver problemas da área de *Smart Farming* com uso de técnicas de processamento de imagem e conceitos de composição de serviços, demandam ainda por iniciativas que deem suporte a construção de novas aplicações. Logo, esse trabalho objetiva discutir a junção desses três temas: composição de serviços, processamento de imagens, e *Smart Farming*.

### 3. Organização do estudo, conceitos e resultados

Nos últimos anos, a agropecuária tem lidado cada vez mais com as mudanças trazidas pela revolução digital proporcionada pela Internet das Coisas [Savvas 2015]. A utilização de conceitos da Internet das Coisas é algo essencial para a consolidação da *Smart Farming*, na qual é habilitada por sistemas agrícolas automatizados, capazes de monitorar eventos das propriedades e controlar os dispositivos implantados de acordo com os dados coletados [Ryu et al. 2015].

No domínio das aplicações para *Smart Farming* pode-se destacar aplicações que utilizam técnicas de processamento de imagem para tomada de decisões. As imagens capturadas podem ser processadas, podendo ser utilizadas para contagem de objetos em um ambiente ou na detecção de pragas em uma lavoura, por exemplo.

Nesse contexto, esse trabalho investiga o uso de aplicações para *Smart Farming* que utilizam técnicas de processamento de imagens de modo a permitir reutilização de componentes de *software* através do emprego dos conceitos de computação orientada a serviços. Sendo assim, foi utilizada uma metodologia exploratória para selecionar os conceitos de composição de serviços dentro de *Smart Farming*, linguagem de modelagem e, linguagens e técnicas para o processamento de imagem digital, conforme as discussões das subseções a seguir. Posteriormente, os resultados discutidos nessa seção são utilizados para a implementação de uma composição de serviços na qual estudos de caso são explorados na Seção 4.

#### 3.1. Composição de serviços

A primeira fase deste trabalho foi buscar conhecer os principais termos para utilização de serviços *web* e sua composição em novos serviços. Logo, duas maneiras principais de de-

envolver serviços *web* foram identificadas: serviços baseado em SOAP e serviços RESTful [Pautasso et al. 2008]. Serviços RESTful têm se mostrando um padrão mais adequado para *Smart Farming* por apresentar uma sintaxe mais enxuta para troca de mensagem e por permitir menor acoplamento, garantindo assim mais flexibilidade nas alterações e reutilização de serviços entre diferentes aplicações [Sheng et al. 2014].

Ao construir os serviços, é necessário definir a sequência de atividades da composição, ou seja, a troca de mensagens entre os serviços. Existem duas maneiras para descrever um processo de negócios: orquestração e coreografia [Decker et al. 2008]. A orquestração de serviços representa um único processo de negócios executável que coordena a interação entre os diferentes serviços, descrevendo um fluxo da perspectiva e sob o controle de um único ponto de extremidade. A coreografia é normalmente associada às interações que ocorrem entre vários serviços *web* em vez de um processo de negócios específico que uma única parte executa.

Para modelar uma composição de serviços, pode-se utilizar uma linguagem de modelagem para composição de serviços, dentre as principais pode-se citar: *Business Process Execution Language para Web Services* (BPEL) [Decker et al. 2008] e o *Business Process Modeling Notation* (BPMN) [OMG 2011]. A linguagem BPEL é um padrão bastante citado na literatura para implementação de processos de negócios com a utilização de tecnologias de serviços *web*. Por outro lado, o BPMN tem alcançado um alto nível de adoção entre analistas de negócios e arquitetos de sistemas como linguagem para a definição de esquemas de processos de negócios para implementação subsequente. Ambas suportam a modelagem de orquestração de serviços.

### 3.2. Processamento de imagem

Como discutido anteriormente, uso de composições de serviços desempenha um papel significativo na consolidação da Internet das Coisas [Alsaryrah et al. 2019] e na implantação da *Smart Farming* [Rupanagudi et al. 2015]. A maioria das aplicações no contexto de *Smart Farming* fazem uso de informações capturadas por sensores e imagens capturadas por equipamentos específicos, como satélites e drones. Tais conjuntos de dados necessitam ser previamente processados e, em seguida, integrados para gerar a solução requisitada pelos produtores [Bhange and Hingoliwala 2015].

A escolha adequada da linguagem de programação para implementação das técnicas de processamento de imagem no contexto de *Smart Farming* é crucial devido tanto a necessidade de uma linguagem e ambiente robustos quanto pela capacidade de integração com diversos sistemas e coisas. Diante disso, esse trabalho buscou avaliar qual a linguagem de programação poderia ser mais adequada neste contexto. Como primeiro passo, foi realizada uma revisão na literatura a procura por trabalhos que lidam com processamento de imagem e que discutem a implementação das suas abordagens. Diante dessa revisão, foi obtido que as linguagens mais utilizadas para processamento de imagem são a ferramenta de prototipagem Matlab e as linguagens Java e Python.

Geralmente a resolução de problemas de processamento de imagens são implementados utilizando a ferramenta Matlab. Isso ocorre porque Matlab é um ambiente de desenvolvimento rápido e permite a construção de soluções para problemas quase exatamente como elas são definidas matematicamente. Entretanto, o Matlab é um *software* proprietário com alto custo de licença e com limitada capacidade de integração com outros sistemas. Sendo assim, programas em Matlab não são adequados para construção de

aplicações no contexto da *Smart Farming*, que requer a criação de processos de negócio flexíveis e dinâmicos.

Diante disso, a principal dificuldade em lidar com aplicações de processamento de imagem dentro do contexto da *Smart Farming* é identificar uma arquitetura que esteja em conformidade com uso de técnicas de processamento de imagem de forma a permitir separar a funcionalidade de um programa em módulos independentes e intercambiáveis. Tal arquitetura deve ser capaz de atender a novos requisitos e facilitar a manutenção futura. Por esse motivo, as linguagens Java e Python são linguagens que vêm sendo utilizadas tanto para tratar de processamento de imagem quanto para implementar composições de serviços [Cokelaer et al. 2013, Van der Walt et al. 2014, Rupanagudi et al. 2015, Ray 2017].

Como essas duas linguagens contam com diversas bibliotecas que auxiliam na construção de aplicações que necessitam de técnicas de processamento de imagem, foi realizada a comparação entre elas. As principais métricas definidas para realizar a comparação entre as linguagens foi o tempo de processamento, representado através do número de linhas de códigos, e o tempo médio de execução em milissegundos, sendo esse o valor obtido através da média de execução em 100 vezes.

Em seguida, foram definidos os algoritmos que deveriam ser implementados utilizando algoritmos fornecidos pelas bibliotecas das linguagens. Os algoritmos foram filtro de escala de cinza, filtro de negativo, filtro de brilho, rotação e segmentação. Todos esses algoritmos são fornecidos pelas bibliotecas das linguagens Java e Python. Os resultados são mostrados na Tabela 1. As métricas com melhor valor recebeu valor 1 e os outros receberam valores relativos a ele. Por exemplo, para realizar o processo de segmentação, o número de linhas de código em Java é 2.66 vezes maior do que o número de linhas necessárias em Python; o mesmo vale para tempo de execução, onde Python é 42% mais rápido que Java para realizar a segmentação.

**Tabela 1. Resultados da execução dos códigos de acordo com as métricas de linhas de código e tempo de execução(milissegundos).**

| Técnicas             | Métricas |        |       |        |
|----------------------|----------|--------|-------|--------|
|                      | Linhas   |        | Tempo |        |
|                      | Java     | Python | Java  | Python |
| <b>Tons de cinza</b> | 2,5      | 1      | 1,35  | 1      |
| <b>Negativo</b>      | 4        | 1      | 1,07  | 1      |
| <b>Brilho</b>        | 3        | 1      | 1,59  | 1      |
| <b>Rotação</b>       | 1        | 1      | 1     | 2,18   |
| <b>Segmentação</b>   | 2,66     | 1      | 1,42  | 1      |

### 3.3. Resultados

Diante do exposto, a nossa hipótese é que composição de serviços se mostra uma solução viável para criação de aplicações de processamento de imagem dentro do contexto da *Smart Farming* por promover a ideia de montar componentes em uma rede de serviços fracamente acoplada, na qual tais serviços são disponibilizados por provedores de serviços através da *web*. Sendo assim, nesse trabalho optamos por utilizar composição de serviços do tipo RESTful com o uso de um orquestrador. Já para a modelagem, utilizamos um

modelo baseado em BPMN, uma vez que é o padrão **de facto** para modelagem de processo de negócios, logo essa notação se torna também indicada para modelagem de problemas na área de *Smart Farming*.

De acordo com os resultados da Tabela 1 é possível notar que a linguagem Python possui um desempenho superior em relação ao tempo de processamento em comparação com a linguagem Java. Já em relação a métrica de linhas de código, novamente Python possui vantagem em relação a Java. Sendo assim, Python se mostra uma linguagem adequada para lidar com processamento de imagem dentro do contexto da *Smart Farming*. Entretanto, vale ressaltar que ao lidar com composições de serviços, a linguagem de implementação para os serviços podem ser diferentes, ou seja, não influencia na lógica de execução da composição.

#### 4. Experimento e Discussão

O objetivo desse experimento foi mostrar o uso da composição de serviços para resolver um problema de processamento de imagem dentro de contexto da *Smart Farming*. A tarefa de identificar e contar certos tipos de objetos em uma fazenda é uma tarefa árdua, mas ao aplicar técnicas de processamento de imagens, esse processo pode ser realizado com pouca intervenção humana, garantindo maior confiabilidade nos resultados. Nesse contexto, dois problemas foram selecionados:

- Contagem de bovinos: em uma fazenda que trabalha com pecuária, o desaparecimento de bovinos podem ocorrer devido a fatores diversos, tais como fuga, furto e morte. O objetivo da aplicação é contar quantos animais estão em uma certa área, permitindo que um sistema de *Smart Farming* integrado possa gerar um alerta quanto a quantidade de animais for inferior ao número do plantel atual.
- Contagem de moscas-brancas: em uma lavoura, a mosca-branca é uma praga que causa grandes danos, tanto danos diretos quanto indiretos por atuar como vetor de vírus fitopatogênicos [de OLIVEIRA and Lima 2006]. Para um maior controle sobre essa praga, existe a necessidade da contagem da quantidade de moscas-brancas que estão em uma folha de uma planta. Logo, o objetivo da aplicação é contar quantas moscas-brancas estão em uma certa folha, permitindo assim que um sistema de *Smart Farming* integrado recomende o manejo adequado.

Nesse processo de identificação e contagem de objetos, inicialmente foram identificadas as funcionalidades exigidas e, em seguida, foi modelado a composição de serviços baseada em BPMN, Figura 1. Como se observa, quatro atividades são requeridas: conversão da imagem de entrada, segmentação, erosão, e determinação da área de interesse (*bounding box*). Cada atividade é responsável por uma parte do processamento o que garante uma clara separação entre os componentes.

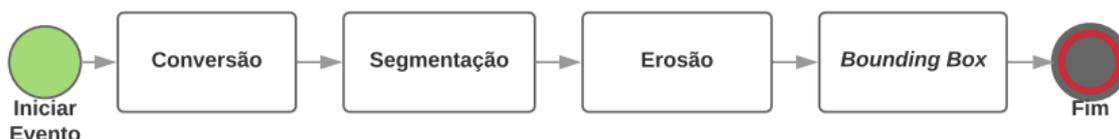


Figura 1. Modelo para identificação e contagem de objetos utilizando composição de serviços de processamento de imagem.

Em seguida, os serviços concretos que fornecem as funcionalidades requisitadas devem ser selecionados e vinculados a cada atividade na composição. Nesse sentido, vários serviços foram construídos utilizando o padrão RESTful e a linguagem Python, sendo implantados em um servidor *web*. Cada serviço é responsável por uma parte do processamento, garantindo assim o baixo acoplamento. Para facilitar a criação dos serviços de processamento de imagens utilizados neste trabalho, foi utilizada a biblioteca OpenCV [OpenCV 2019] que é amplamente adotada na literatura e contém uma quantidade significativa de algoritmos de processamento de imagens e algoritmos de visão computacional, como segmentação, classificação, reconhecimento de padrão e rastreamento de objetos.

Uma arquitetura baseada em serviços permite incrementar o reuso de serviços entre várias composições, uma vez que um serviço com funcionalidades requeridas por várias composições pode ser utilizado de forma compartilhada. Outra opção é que um serviço com funcionalidade requerida por várias composições possa ter múltiplas instâncias, sendo uma para cada composição. Além disso, a manutenção das aplicações é bastante simplificada, uma vez que um serviço por ser substituído por outro, caso seja necessário um maior poder de processamento ou simplesmente para mudar a funcionalidade da aplicação. Nesses casos, essa substituição de serviços em uma composição é transparente ao usuário, ou seja, o usuário considera apenas um serviço único e não um conjunto de serviços distintos que podem ser substituídos.

Depois da criação e disponibilização dos serviços na *web*, foi implementada a lógica do modelo da Figura 1 para cada um dos dois problemas discutido anteriormente. Esses dois problemas são independentes um do outro, mas possuem algumas características em comum, dentre elas o reconhecimento e a contagem de objetos. Sendo assim, foram codificados dois experimentos, onde ambos utilizam o mesmo modelo e um grupo de serviços é compartilhado. A seguir os dois experimentos são discutidos em detalhes.

#### 4.1. Contagem de bovinos

O primeiro experimento lida com contagem de vacas em uma imagem capturada por um drone e, então, submetida para o serviço de contagem de bovinos. Essa aplicação foi implementada sobre a lógica do modelo da Figura 1, ou seja, foi construída como uma composição de serviços que implementam um conjunto de técnicas de processamento de imagem de forma orquestrada para obter o resultado desejado.

Dado o modelo da Figura 1, o primeiro serviço a ser executado é o *Serviço Conversão* que é responsável por receber a imagem e iniciar a execução da lógica da composição. No experimento, esse serviço recebe uma imagem RGB (Figura 2(a)) como entrada e produz uma imagem em formato HSV, uma vez que para realizar a segmentação é requerida uma imagem em tal formato para ser processada. A saída deste serviço é mostrada na Figura 2(b), onde essa imagem é enviada para o próximo serviço no fluxo de execução.

Em seguida, a imagem convertida é submetida para o *Serviço Segmentação* que tem por responsabilidade realizar a segmentação da imagem com base em suas cores. A operação de segmentação consiste em dividir os *pixels* da imagem em duas classes, objetos de primeiro plano e objetos de segundo plano. Nesse experimento os *pixels* buscados são os brancos por serem tratados como *pixels* pertencentes ao objeto vaca. Ao encontrar um *pixel* branco (definido por um limiar) a função de segmentação da biblioteca rotula



**Figura 2. (a) A imagem original capturada e submetida à composição de serviços. (b) Imagem depois de realizada a conversão para o formato HSV.**

esse ponto como objeto vaca, caso contrário o *pixel* é tratado como objeto de segundo plano. A imagem com a operação de segmentação é mostrada na Figura 3.

O terceiro serviço do modelo, chamado de *Serviço Erosão*, é usado para realizar operações morfológicas na imagem. Como temos uma imagem que representa um pasto, os ruídos na imagem podem levar a resultados incorretos, portanto, usamos a operação morfológica chamada erosão para remover os elementos que não representam os bovinos, ou seja, artefatos menores do que um tamanho definido em um limiar. Na imagem da Figura 3 existem várias regiões com pequenas manchas, que não têm tamanho ideal de uma vaca, ou seja, todas as pequenas manchas encontradas na imagem são tratadas como ruídos. Sendo assim, o *Serviço Erosão* recebe uma imagem segmentada e busca a diminuição dos ruídos produzindo uma nova representação. O resultado desse serviço é mostrado na imagem da Figura 4(a), em seguida, essa imagem é enviada para o próximo serviço no fluxo de execução.



**Figura 3. Imagem segmentada.**

O quarto e último serviço a ser chamado da composição para realizar aplicação é o *Serviço Bounding Box*. Esse serviço recebe a imagem produzida pelo *Serviço Erosão* e é responsável por destacar os objetos encontrados na imagem. Nesse experimento, os objetos encontrados em uma imagem são marcados com círculos, o que permite a contagem do bovinos encontrados. Ao executar o serviço *Serviço Bounding Box*, o fluxo

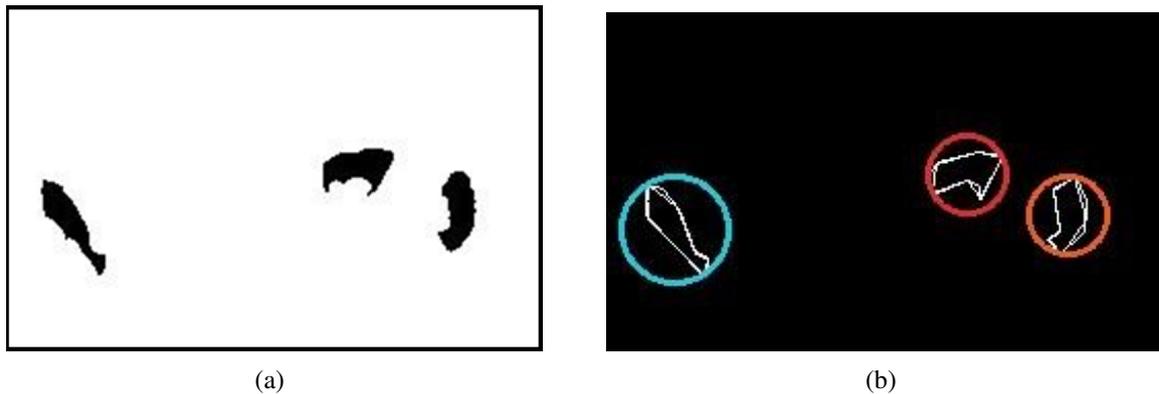


Figura 4. (a) Remoção de ruídos e (b) Vacas detectadas.

de execução da lógica da composição é finalizado, sendo produzido uma saída como mostra a Figura 4(b). Como pode ser observado na Figura 4(b), três objetos foram circulares, ou seja, três bovinos foram encontrados e destacados. Logo, pode-se perceber que a composição de serviços foi capaz de identificar e contar corretamente o número de animais em uma imagem.

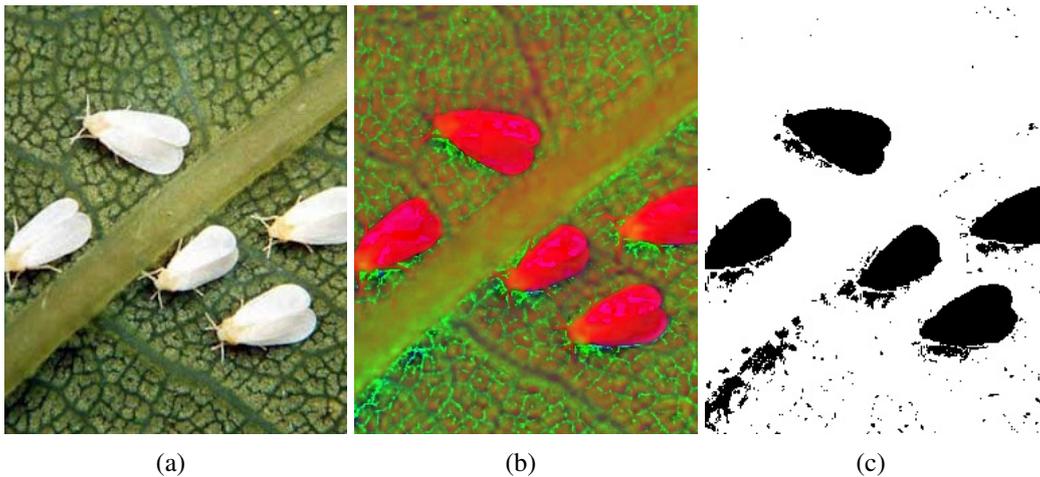
#### 4.2. Contagem de moscas-brancas

Para este segundo experimento foi utilizado o problema de contagem de moscas-brancas em folhas de uma plantação. Diferente do primeiro experimento, que lida com a área de pecuária, esse experimento trata do controle de pragas em lavoura. Entretanto, como ambos problemas possuem características em comum, o mesmo modelo de composição é utilizado. A diferença entre as duas soluções está na seleção dos serviços adequados para cada atividade da composição.

Nos dois experimentos, o serviço para lidar com a segmentação deve ter uma implementação diferente para cada experimento, que permita a escolha de um diferente limiar, ou ainda, outro método de segmentação. Dado o modelo da Figura 1, basta substituir a implementação da segunda atividade por um outra implementação do *Serviço Segmentação*, inclusive esse serviço pode ser disponibilizado por terceiros. Dessa forma, consegue-se garantir o baixo acoplamento entre os serviços da composição visto que com a mesma arquitetura e com diferentes imagens de entrada é possível alcançar o resultado desejado apenas com a substituição de serviços.

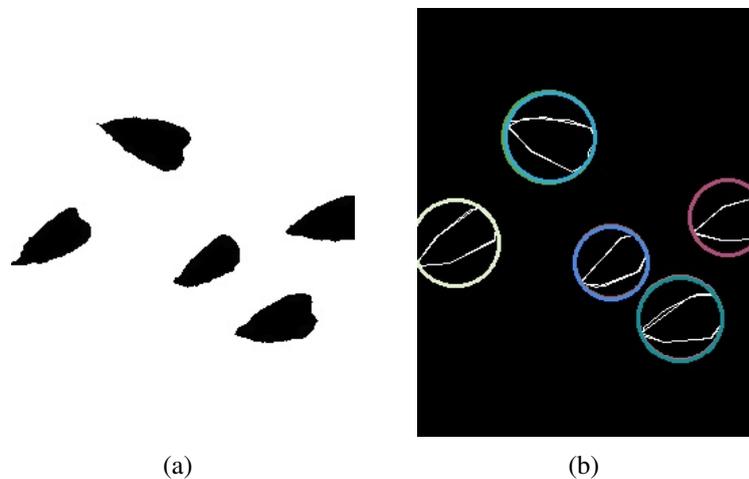
De modo semelhante, seguindo a lógica do modelo da Figura 1, nesse experimento o primeiro serviço a ser executado é o *Serviço Conversão*. A Figura 5(a) mostra a imagem de entrada utilizada para a contagem de moscas-brancas em uma folha. Ao receber a imagem, o *Serviço Conversão* produz uma imagem em formato HSV, como mostra a Figura 5(b). Mesmo os dois experimentos lidando com imagens diferentes, a mesma implementação do *Serviço Conversão* é compartilhada nas duas composições.

A imagem no formato HSV é submetida para o *Serviço Segmentação* para realizar a segmentação da imagem. Nesse caso, uma outra implementação para o *Serviço Segmentação* foi selecionada, dessa vez, utilizando o limiar automático de imagem por meio do método de Otsu [Otsu 1979]. O resultado da operação de segmentação é mostrada na Figura 5(c).



**Figura 5. (a) A imagem original capturada e submetida à composição de serviços. (b) Imagem depois de realizada a conversão para o formato HSV. (c) Imagem segmentada.**

Em seguida, o *Serviço Segmentação* envia a imagem ao *Serviço Erosão* para minimizar os ruídos da imagem segmentada. O resultado desse serviço é mostrado na imagem da Figura 6(a). Por fim, o *Serviço Bounding Box* recebe a imagem e realiza o processo de marcar os objetos encontrados, como mostra a Figura 6(b).



**Figura 6. (a) Remoção de ruídos e (b) Moscas-brancas detectadas.**

Nesse segundo experimento, as implementações do *Serviço Erosão* e *Serviço Bounding Box* são as mesmas do primeiro experimento, logo esses serviços são compartilhados entre as duas aplicações. Como se pode observar, na imagem da Figura 6(b) cinco objetos foram circulado, ou seja, cinco moscas-brancas foram encontradas e destacadas. Logo, pode-se perceber que o mesmo modelo de composição de serviços pode ser utilizado para resolver dois problemas distintos da *Smart Farming*, ou seja, identificar e contar o número de objetos tanto na imagem para identificação de bovinos quanto na imagem de moscas-brancas. Vale ressaltar que essa composição pode ser facilmente integrada com outras aplicações, construindo sistemas mais complexos.

## 5. Conclusões e Trabalhos Futuro

O estudo do emprego da computação orientada a serviços no desenvolvimento de aplicações de processamento de imagens é muito importante e pode ser utilizada de forma interdisciplinar, com grande utilidade para negócios da *Smart Farming* e outras áreas. Pensando nisso, foi criada uma aplicação baseada em composição de serviços para identificação e contagem de objetos em imagens capturadas no ambiente rural e submetidas a uma aplicação disponibilizada via *web*. O maior desafio foi definir uma aplicação de processamento de imagens que pudesse ser particionada utilizando uma arquitetura orientada a serviços. Para isso, o modelo de negócio da aplicação foi implementado de forma que não só o desenvolvedor possa utilizar os serviços que compõe a aplicação, mas também que outras aplicações possam reutilizar os serviços.

Inicialmente foi realizado um estudo sobre a modelagem e codificação de serviços *web* seguindo as melhores práticas da literatura. Em seguida, os serviços foram codificados e implantados em servidores, de forma a serem disponibilizados na *web*. Para mostrar a viabilidade para uso de composições de serviços de processamento de imagem, dois experimentos foram criados e testados. Vale ressaltar que ambos experimentos lidam com problemas distintos, ou seja, imagem de bovinos e moscas-brancas, mas compartilham vários serviços em comum, inclusive a mesma arquitetura. Os resultados mostraram a viabilidade e a facilidade para compor serviços na construção de aplicações dentro do contexto da *Smart Farming*.

Como trabalho futuro, pretendemos refinar as técnicas de processamento de imagem executadas pelos serviços e aprimorar as soluções apresentadas de modo a lidar com problemas típicos de visão computacional, tais como, oclusões, sombreamentos e segmentação de objetos em sobreposição. Além disso, os serviços serão implantados em recursos fornecidos por provedores de ambiente de nuvem a fim de aumentar o poder computacional dos serviços, bem como sua disponibilidade e escalabilidade.

## Referências

- Alonso, G., Casati, F., Kuno, H., and Machiraju, V. (2004). *Web services: Concepts, Architectures and Applications*. Springer-Verlag Berlin Heidelberg.
- Alsaryrah, O., Mashal, I., and Chung, T.-Y. (2019). A fast iot service composition scheme for energy efficient qos services. In *Proceedings of the 2019 7th International Conference on Computer and Communications Management*, pages 231–237. ACM.
- Autili, M., Inverardi, P., and Tivoli, M. (2014). Choreos: large scale choreographies for the future internet. In *2014 Software Evolution Week-IEEE Conference on Software Maintenance, Reengineering, and Reverse Engineering (CSMR-WCRE)*, pages 391–394. IEEE.
- Baryannis, G., Danylevych, O., Karastoyanova, D., Kritikos, K., Leitner, P., Rosenberg, F., and Wetzstein, B. (2010). Service composition. In Papazoglou, M., Pohl, K., Parkin, M., and Metzger, A., editors, *Service Research Challenges and Solutions for the Future Internet*, volume 6500 of *Lecture Notes in Computer Science*, pages 55–84. Springer Berlin Heidelberg.
- Bertolino, A., De Angelis, G., Polini, A., and Sabetta, A. (2011). Trends and research issues in SOA validation. *Performance and Dependability in Service Computing: Concepts, Techniques and Research Directions*, page 98.

- Bhange, M. and Hingoliwala, H. (2015). Smart farming: Pomegranate disease detection using image processing. *Procedia Computer Science*, 58:280–288.
- Çakır, Y., Kırıcı, M., Güneş, E. O., and Üstündağ, B. B. (2013). Detection of oranges in outdoor conditions. In *2013 Second International Conference on Agro-Geoinformatics (Agro-Geoinformatics)*, pages 500–503. IEEE.
- Cardellini, V., Casalicchio, E., Grassi, V., Iannucci, S., Presti, F. L., and Mirandola, R. (2012). Moses: A framework for QoS driven runtime adaptation of service-oriented systems. *Software Engineering, IEEE Transactions on*, 38(5):1138–1159.
- Cokelaer, T., Pultz, D., Harder, L. M., Serra-Musach, J., and Saez-Rodriguez, J. (2013). Bioservices: a common python package to access biological web services programmatically. *Bioinformatics*, 29(24):3241–3242.
- de OLIVEIRA, M. and Lima, L. (2006). Moscas-brancas na cultura da mandioca no brasil. *Embrapa Recursos Genéticos e Biotecnologia-Documentos (INFOTECA-E)*.
- Decker, G., Kopp, O., Leymann, F., Pfitzner, K., and Weske, M. (2008). Modeling service choreographies using bpmn and bpel4chor. In *Advanced Information Systems Engineering*, pages 79–93. Springer.
- Fan, Y., Wang, T., Qiu, Z., Peng, J., Zhang, C., and He, Y. (2017). Fast detection of striped stem-borer (chilo suppressalis walker) infested rice seedling based on visible/near-infrared hyperspectral imaging system. *Sensors*, 17(11):2470.
- Furtado, T., Francesquini, E., Lago, N., and Kon, F. (2014). Towards an enactment engine for dynamically reconfigurable and scalable choreographies. In *Services (SERVICES), 2014 IEEE World Congress on*, pages 325–332. IEEE.
- Honda, B. and JORGE, L. d. C. (2013). Computação aplicada à agricultura de precisão. *Embrapa Instrumentação-Artigo em periódico indexado (ALICE)*.
- Issarny, V., Georgantas, N., Hachem, S., Zarras, A., Vassiliadis, P., Autili, M., Gerosa, M. A., and Hamida, A. B. (2011). Service-oriented middleware for the future internet: state of the art and research directions. *Journal of Internet Services and Applications*, 2(1):23–45.
- Jhuria, M., Kumar, A., and Borse, R. (2013). Image processing for smart farming: Detection of disease and fruit grading. In *2013 IEEE Second International Conference on Image Information Processing (ICIIP-2013)*, pages 521–526. IEEE.
- OMG (2011). Documents Associated with Business Process Model and Notation (BPMN) Version 2.0. <http://www.omg.org/spec/BPMN/2.0/>.
- OpenCV (2019). Open Source Computer Vision Library 4.1.1. <https://opencv.org>.
- Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66.
- Pautasso, C., Zimmermann, O., and Leymann, F. (2008). Restful web services vs. big web services: making the right architectural decision. In *Proceedings of the 17th international conference on World Wide Web*, pages 805–814. ACM.

- Ray, P. P. (2017). Internet of things for smart agriculture: Technologies, practices and future direction. *Journal of Ambient Intelligence and Smart Environments*, 9(4):395–420.
- Rupanagudi, S. R., Ranjani, B., Nagaraj, P., Bhat, V. G., and Thippeswamy, G. (2015). A novel cloud computing based smart farming system for early detection of borer insects in tomatoes. In *2015 International Conference on Communication, Information & Computing Technology (ICCICT)*, pages 1–6. IEEE.
- Ryu, M., Yun, J., Miao, T., Ahn, I.-Y., Choi, S.-C., and Kim, J. (2015). Design and implementation of a connected farm for smart farming system. In *2015 IEEE SENSORS*, pages 1–4. IEEE.
- Savvas, A. (2015). Farming industry must embrace the internet of things to grow enough food. *Techworld*.
- Sheng, Q. Z., Qiao, X., Vasilakos, A. V., Szabo, C., Bourne, S., and Xu, X. (2014). Web services composition: A decades overview. *Information Sciences*, 280:218–238.
- Van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E., and Yu, T. (2014). scikit-image: image processing in python. *PeerJ*, 2:e453.
- Vieira, G. D. S., Soares, F. A. A., De Lima, J. C., Do Nascimento, H. A., Laureano, G. T., Da Costa, R. M., Ferreira, J. C., and Rodrigues, W. G. (2019). A disparity computation framework. In *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*, volume 2, pages 634–639. IEEE.
- W3C (2002). Web services description requirements. <https://www.w3.org/TR/2002/WD-ws-desc-reqs-20021028/ws-desc-reqs.pdf>.
- Yang, C., Prasher, S., Landry, J., Perret, J., Ramaswamy, H., et al. (2000). Recognition of weeds with image processing and their use with fuzzy logic for precision farming. *Canadian Agricultural Engineering*, 42(4):195–200.
- Yang, L., Dickinson, J., Wu, Q. J., and Lang, S. (2007). A fruit recognition method for automatic harvesting. In *2007 14th International Conference on Mechatronics and Machine Vision in Practice*, pages 152–157. IEEE.