



Visão computacional e reconhecimento de imagem

Apresentação dos aspectos fundamentais de visão computacional e do reconhecimento de imagens.

Prof. Sérgio Assunção Monteiro

Propósito

A inteligência artificial proporciona grandes avanços na forma das pessoas interagirem, como a área de visão computacional, que permite ter acesso a diagnósticos médicos mais precisos, ou mesmo transações financeiras mais seguras utilizando QR Codes. Adquirir conhecimento sobre essa área aumenta as possibilidades de você conseguir destaque no mercado.

Preparação

Todos os nossos exemplos foram implementados no Google Colab. Portanto, você precisa criar uma conta de gmail e acessar o site colab.research.google.com.

Objetivos

- Reconhecer os conceitos de visão computacional e reconhecimento de imagens.
- Analisar os frameworks para visão computacional.
- Reconhecer o processo de detecção e reconhecimento de imagem.
- Identificar as aplicações de visão computacional e do reconhecimento de imagens.

Introdução

Nos dias de hoje, a maioria de nós utiliza aplicativos para pagar contas. Basicamente, fazemos a leitura do código de barras e todas as informações já são carregadas. Tudo que precisamos fazer é verificar se os dados estão corretos e efetuar o pagamento. Esse é apenas um dos diversos exemplos em que podemos aplicar as técnicas de visão computacional e reconhecimento de imagens.

A teoria por trás dessas técnicas é bastante sofisticada e envolve álgebra linear, processamento de sinais, trigonometria, anatomia do sistema ocular, estatística, física, entre outras tantas áreas. Na prática, utilizamos ferramentas que facilitam o nosso trabalho. Assim, simplesmente fazemos chamadas para funções que executam as tarefas que precisamos, sem a necessidade de entrarmos em detalhes excessivos.

Ao longo deste estudo, teremos a oportunidade de conhecer um pouco mais sobre essas teorias e, principalmente, vamos ver alguns exemplos práticos muito importantes. Tente executar cada um deles para criar segurança. Além disso, dê o passo seguinte: realize modificações nos exemplos e tente acrescentar novas funcionalidades. Essa área é fascinante e tem muito ainda para crescer. Então, que “comecem os jogos”!



Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

Aspectos essenciais

Neste vídeo, vamos entender o que significa cada um dos aspectos essenciais de visão computacional e reconhecimento.



Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

Hoje em dia, interagimos com sistemas cada vez mais “espertos”. Muito provavelmente, você já deve ter feito consultas na internet simplesmente perguntando: “Onde posso comprar uma pizza mais perto da minha casa?” Ou, ainda, já deve ter utilizado uma rede social com fotos em que pessoas conhecidas foram identificadas e sugeridas para serem conectadas a você.

Apesar de toda a tecnologia envolvida nesses cenários, a parte mais importante é o que está por trás deles: algoritmos de inteligência artificial.

E aí vem a pergunta: afinal de contas, o que é inteligência artificial?

Uma parte dessa pergunta é fácil de ser respondida: o termo artificial está relacionado a tudo que passa por um processo de transformação, ou seja, que não é encontrado no seu estado natural. No nosso caso, estamos tratando de algoritmos, que são sequências lógicas que nos permitem programar equipamentos eletrônicos.

Agora, vamos para a parte mais complexa da pergunta: o que é inteligência? A inteligência está relacionada à capacidade de generalizar aprendizados para resolver problemas.

Para ficar mais fácil de entender, vamos fazer uma analogia: ao observarmos um aluno do primeiro grau estudando sobre como resolver um problema de matemática, sabemos que a primeira parte do aprendizado está relacionada a conhecer fórmulas e aplicá-las — talvez agora você esteja se lembrando dos bons períodos da sua época de escola! Na sequência, esse estudante passa a aplicar o conhecimento que adquiriu em situações similares, mas com outros dados. Ou seja, passa a generalizar o aprendizado. Com o tempo, o estudante combina diversos conhecimentos para resolver problemas mais complexos.

A inteligência artificial tenta reproduzir as características da inteligência humana, por meio de técnicas que combinam modelos computacionais e algoritmos que estão em constante processo de evolução. Uma das aplicações mais interessantes nesse contexto é a de visão computacional.

Aqui, nós vamos explorar com muitos detalhes esse assunto. Por enquanto, basta entendermos que ele está relacionado à capacidade dos sistemas computacionais coletarem dados de imagem e interpretá-los.

Além da visão computacional, temos diversas outras aplicações da inteligência artificial que abrangem sistemas especialistas, processamento de linguagem natural e reconhecimento de fala. De um modo geral, podemos considerar esses sistemas como os que agem como humanos e que constroem estruturas lógicas como humanos. No entanto, na abordagem ideal, queremos obter sistemas que utilizam construções lógicas racionais para tomar decisões.

Aí vem a próxima pergunta: o que é o pensamento racional?

O pensamento racional é essencialmente um processo que nos permite tirar conclusões sensatas baseadas em fatos, lógica e dados. Ele tem por objetivo resolver problemas para alcançar um objetivo.

Todos esses aspectos de que tratamos serão usados mais adiante. A seguir, antes de entrarmos na visão computacional, abordaremos brevemente dois aspectos essenciais: aprendizado de máquina e aprendizado profundo.

Aprendizado de máquina

É uma das áreas da inteligência artificial. Sua principal característica é utilizar modelos computacionais que extraem características dos dados mediante um processo que permite realizar generalizações. Daí vem o termo “aprendizado”, ou seja, identificar estruturas lógicas nos dados para tomar decisões sobre circunstâncias semelhantes. Dessa forma, os algoritmos permitem reproduzir comportamentos que sejam coerentes com os dados que foram analisados.

É muito importante que saibamos diferenciar um algoritmo tradicional de um de aprendizado de máquina.

Algoritmos tradicionais

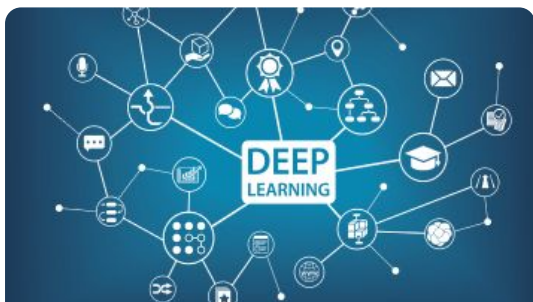
Nesses algoritmos, utilizamos comandos que definem como os dados serão processados. Esses comandos são uma combinação de condições e laços de repetição que são estabelecidos previamente. Portanto, temos a expectativa de comportamentos bem controlados e que podem ser testados.

Algoritmos de aprendizado de máquina

Com esses algoritmos nós trabalhamos de uma forma bem diferente. Utilizamos modelos computacionais que combinam arquitetura e estratégias de processamento dos dados. Esses algoritmos permitem realizar classificações ou previsões e descobrir informações importantes.

Aprendizado profundo

É uma especialização do aprendizado de máquina. Algoritmos desse tipo são caracterizados, principalmente, por modelos computacionais complexos que utilizam camadas lógicas para capturar as relações entre os dados. Esses modelos são uma combinação das chamadas redes neurais e de modelos especializados.



Infográfico de aprendizagem profunda.

Os modelos de redes neurais de aprendizado profundo mais conhecidos são as **redes de convolução** (CNNs) e as **redes de recorrência** (RNNs). As CNNs são utilizadas, principalmente, para aplicações de visão computacional. Já as RNNs são utilizadas para aplicações de previsões de séries temporais e processamento de linguagem natural.

Aspectos da visão computacional

Neste vídeo, falaremos sobre alguns aspectos da visão computacional destacando alguns pontos como definição e levantamento de dados, dentre outros.



Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

Definição de visão computacional

A visão computacional combina diversas técnicas que permitem que os dispositivos eletrônicos funcionem de forma semelhante aos olhos humanos. Para que seja possível utilizar aplicações práticas nessa área, precisamos levar em consideração os aspectos de hardware, como o de software. Isso ocorre porque é necessário haver a coleta dos dados de imagem — que é responsabilidade do hardware — e, posteriormente, a identificação de relações entre esses dados — que é de responsabilidade dos algoritmos (software).



Reflexão

Hoje em dia, é bastante comum utilizarmos aplicativos de visão computacional em telefones celulares e outros dispositivos eletrônicos, voltados tanto para entretenimento como para segurança. Isso só é possível pelo avanço da tecnologia e do desenvolvimento dos algoritmos de aprendizado de máquina e, posteriormente, de aprendizado profundo, que foram essenciais para viabilizar o uso de visão computacional para tantas aplicações.

A ideia principal é a de replicar partes da complexidade do sistema de visão humana, de maneira a permitir que os sistemas computacionais identifiquem e processem objetos em imagens e vídeos de uma forma semelhante ao que ocorre com a visão dos humanos e, em alguns casos, de forma até mais sofisticada.

É importante entendermos que a visão computacional faz parte da inteligência artificial (IA). As imagens digitais, vídeos e outras entradas visuais são coletadas e, posteriormente, transformadas em dados, para que possam ser processados. A partir desse processamento, as informações podem ser usadas como suporte para tomada de decisões ou, ainda, para fazer recomendações.



Atenção

Nesse ponto, precisamos fazer algumas considerações éticas, pois esses sistemas são instrumentos muito sofisticados e, sem o devido controle, podem ser usados de formas inadequadas e perigosas. Por exemplo, os sistemas podem ser treinados para identificar pessoas que se opõem a um regime político.

Sob o ponto de vista técnico, os algoritmos de visão computacional são similares aos demais processos de aprendizado de máquina: combinam modelos computacionais e estratégias de processamento para extrair características relevantes que geram conhecimento. Esse processo permite que os modelos realizem generalizações. A combinação dessas etapas é o que chamamos de aprendizado.

Agora que tivemos uma ideia sobre a visão computacional, vamos abordar um aspecto muito importante: o reconhecimento de imagens.

Definição de reconhecimento de imagens

O reconhecimento de imagem é uma subcategoria da visão computacional. Ela utiliza um conjunto de métodos que nos permite processar uma imagem, para que possamos detectar e analisar características específicas. A partir dessa análise, podemos realizar a automação de tarefas específicas. Um bom exemplo de aplicação ocorre no controle de qualidade na fabricação de produtos industriais que devem atender a determinados critérios específicos.

Os algoritmos de reconhecimento de imagens também nos permitem identificar lugares, pessoas e objetos dentro de uma imagem. A partir dessa identificação, podemos tirar conclusões e tomar as ações adequadas. Esse processo de reconhecimento pode ser realizado com diferentes níveis de precisão. Por exemplo, quando analisamos imagens de satélite, podemos monitorar o desmatamento de uma determinada área. À medida que queremos detalhar o que ocorreu com a área, vamos precisar utilizar de técnicas e recursos computacionais mais sofisticados, que combinam estruturas de dados com uma arquitetura de hardware que viabilize realizar uma grande quantidade de cálculos.

Ainda sobre o reconhecimento de imagens, precisamos entender o que podemos realizar com ela. Portanto, temos as seguintes tarefas:

1 Classificação

É a identificação da categoria mais adequada para imagem. Portanto, precisamos fornecer com antecedência um conjunto de classes em que a imagem possa ser classificada. Dessa forma, o algoritmo que aplicarmos será responsável pela identificação correta da classe em que a imagem deve ser classificada. Por exemplo, uma imagem corresponde a um cachorro.

2

Marcação

É uma especialização da tarefa de classificação. O objetivo é reconhecer elementos dentro de uma imagem. Em inglês, é usado o termo tag para expressar marcação que é a base do padrão XML, por exemplo. Um ponto bastante interessante aqui é que podemos ter mais de uma “tag” para a mesma imagem. Por exemplo, uma imagem tem um carro (veículo) que é azul (cor).

3

Deteção

É o processo de localização de objetos ou de características em uma imagem. Quando localizamos o objeto (ou as características), o algoritmo traça uma caixa delimitadora ao redor do objeto em questão.

4

Segmentação

É uma especialização dos algoritmos de detecção. Basicamente, trata-se de um conjunto de técnicas que pode localizar um elemento em uma imagem por meio de suposições a respeito dos elementos da imagem. Por exemplo, podemos usar aspectos da textura dos objetos e materiais para distingui-los.

A seguir, vamos estudar as etapas da visão computacional.

Etapas da visão computacional

Neste vídeo, falaremos sobre algumas etapas da visão computacional, como entrada de dados, pré processamento, dentre outros.



Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

Principais etapas da visão computacional

Nos dias atuais, podemos utilizar as aplicações de visão computacional devido aos diversos trabalhos desenvolvidos na área, que fizeram com que houvesse uma grande evolução. Um dos aspectos essenciais nesse processo evolutivo foi a utilização de uma arquitetura em camadas. Isso tornou viável a concentração na melhoria de cada uma dessas etapas. Por exemplo, antes que a imagem seja fornecida ao modelo para classificação, ela precisa ser pré-processada. Abaixo, apresentamos quais são as etapas da visão computacional:

- Entrada de dados.
- Pré-processamento.
- Seleção de área de interesse.
- Extração de características.
- Predição e reconhecimento.

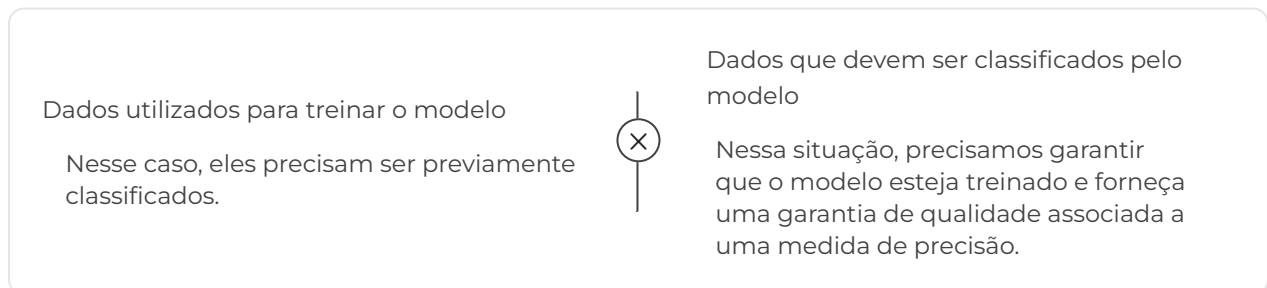
A seguir, vamos analisar cada uma delas.

Entrada de dados

A primeira etapa de um método de visão computacional é a entrada de dados. Isso abrange:

- O meio de coleta dos dados.
- O formato dos arquivos.
- O volume dos dados.
- A velocidade que os dados são gerados e precisam ser processados.
- A garantia de qualidade dos dados.

Além disso, ainda temos duas situações distintas:



Pré-processamento

Depois de coletar os dados, precisamos realizar um trabalho inicial para garantir a sua qualidade. Isso envolve verificar origem, formato, tamanho e o conteúdo. Em algumas situações, precisamos desconsiderá-los. Isso vai depender do tipo de aplicação e como o modelo de visão computacional foi treinado. Há situações em que os modelos realmente estão aptos para trabalhar com dados com má qualidade e, ainda assim, são capazes de realizar classificações. Na maioria dos casos, porém, os modelos precisam de uma garantia para poder realizar o trabalho de classificação.

Seleção de área de interesse

Um aspecto importante é a definição das áreas de interesse dentro da imagem. O objetivo é o de reduzir o ruído e melhorar a precisão dos algoritmos. Essa seleção ocorre na detecção de objetos. Nesse caso, precisamos gerar caixas delimitadoras da área em que estão os objetos. Dessa forma, eles podem ser posteriormente classificados pelo modelo de visão computacional.

Extração de características

Nós realizamos essa etapa na fase de treinamento do modelo. O objetivo é garantir que o modelo seja capaz de generalizar conhecimento. Por exemplo, o modelo deve ser capaz de distinguir os olhos e a boca de um rosto humano.

Predição e reconhecimento

Essa é a etapa na qual realmente vemos o modelo em ação. Aqui, fornecemos imagens que o modelo não conhece. Então, a partir de sua base de conhecimento, ele fará a classificação dessas imagens.

Com isso, fechamos a primeira parte do nosso estudo sobre os conceitos de visão computacional e reconhecimento de imagens. Na sequência, vamos entrar numa parte mais prática por meio de alguns frameworks que nos ajudam a desenvolver aplicações interessantes.

Verificando o aprendizado

Questão 1

As aplicações de inteligência artificial são uma realidade. Um exemplo concreto dessa afirmação é a área de visão computacional. As aplicações nessa área são caracterizadas por algoritmos que utilizam técnicas avançadas para simular o processo de visão do ser humano. Nesse sentido, selecione a opção correta a respeito das técnicas aplicadas para visão computacional:

A

Os algoritmos de aprendizado profundo produzem os melhores resultados.

B

Os algoritmos tradicionais de aprendizado de máquina são os mais adequados.

C

Qualquer algoritmo de inteligência artificial pode ser adaptado.

D

Os algoritmos de aprendizado de máquina tradicionais são mais precisos do que os de aprendizado profundo.

E

Os algoritmos de aprendizado profundo só podem ser utilizados para visão computacional.



A alternativa A está correta.

Os algoritmos de aprendizado profundo foram muito bem-sucedidos em diversas áreas, entre elas a de visão computacional. No caso específico dessa área, esses algoritmos utilizam redes de convolução.

Questão 2

Utilizar um sistema de visão computacional é uma excelente experiência. Isso ocorre pela capacidade dos algoritmos de realizarem operações que seriam muito complexas para uma pessoa e, realmente, gerarem excelentes resultados. Nesse sentido, selecione a opção correta a respeito do reconhecimento de imagens:

A

Uma aplicação tradicional de reconhecimento de imagens abrange a mudança das cores dos itens.

B

Uma importante aplicação é a de determinar a que categoria um item da imagem pertence.

C

Um aspecto que sempre faz parte do reconhecimento de imagens é determinar se a imagem está ou não em movimento.

D

Uma das vantagens de utilizar reconhecimento de imagens é não precisar da intervenção humana para conferir se a classificação está correta.

E

Os algoritmos de aprendizado de máquina profundo só podem ser usados para aplicações de visão computacional.



A alternativa B está correta.

O reconhecimento de imagens é um dos pilares da visão computacional. Por meio dele é possível categorizar imagens, detectar itens, caracterizar padrões e realizar aperfeiçoamentos nas imagens que têm diversas utilidades, desde o entretenimento às aplicações nas áreas de segurança e médica.

Evolução das ferramentas de visão computacional

Neste vídeo, falaremos sobre alguns aspectos a evolução das ferramentas de visão computacional e alguns exemplos importantes.



Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

A evolução da visão computacional: um breve histórico

A visão computacional tem sua origem nas pesquisas neurofisiológicas realizadas em gatos nas décadas de 1950 e 1960. Na época, os pesquisadores estudaram como os neurônios reagiam a diversos tipos de estímulos. Um resultado fundamental desse estudo foi a descoberta de que a visão humana é hierárquica. Basicamente, os neurônios seguem uma linha de processamento na seguinte ordem:

- Detectam recursos simples como bordas das imagens.
- Em seguida, combinam essas bordas para construir formas.
- Finalmente, constroem representações visuais complexas.

A partir desses trabalhos, os pesquisadores começaram a trabalhar no desenvolvimento de modelos computacionais que reproduzissem as estruturas neurológicas humanas. Nesse processo evolucionário, os sistemas de visão computacional também passaram a aplicar uma abordagem hierárquica. Inclusive, o primeiro trabalho de maior relevância nessa área foi publicado em 1959 por dois neurofisiologistas: David Hubel e Torsten Wiesel. O título da publicação (traduzida para o português) é *Campos receptivos de neurônios individuais no córtex estriado do gato*.

Um outro ponto marcante na evolução da visão computacional foi a invenção do aparelho de scanner de imagem digital, em 1959. O scanner permitia transformar imagens em grades de números, ou seja, transformava uma imagem em linguagem binária.

Dessa forma, foi possível manipular as imagens digitalmente, uma vez que essa era uma representação compatível com a forma que os computadores trabalhavam (e ainda trabalham).



Ainda nesse processo histórico foi publicado, em 1963, o trabalho de título (traduzido para o português) *Percepção da máquina de sólidos tridimensionais*, de Lawrence Roberts. Nesse trabalho, ele descreveu o processo de obtenção de informações tridimensionais sobre objetos sólidos a partir de fotografias bidimensionais. Na prática, ele criou uma forma de associar o mundo visual a formas geométricas simples.

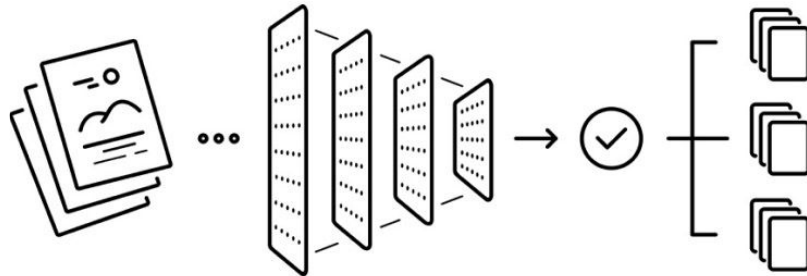
Alguns anos mais tarde, já na década de 1980, um cientista da computação japonês, Kunihiko Fukushima, desenvolveu um modelo computacional capaz de reconhecer padrões. Além disso, esse modelo não sofria prejuízos de qualidade dos resultados quando havia mudanças de posição dos objetos. Esse modelo computacional era uma rede neural que ficou conhecida como Neocognitron.

Alguns anos depois, no final da mesma década, o cientista francês Yann LeCun desenvolveu um modelo computacional que combinava uma arquitetura de camadas especializadas e um algoritmo de aprendizado máquina, que ficou conhecido como LeNet-5. É uma rede neural de aprendizado profundo que utiliza uma função de **convolução**. Esse tipo de modelo computacional é conhecido como redes neurais de convolução,

comumente referenciadas como redes CNN (convolutional neural network). Também são conhecidas como redes convnet.

Convolução

A convolução, de forma bem simples, é uma operação que extrai as principais características de uma imagem.



Classificação de imagens. Aprendizado de máquina e rede neural convolucional.

Ainda nos dias de hoje a rede LeNet-5 é bastante utilizada, tanto para reconhecer imagens como para explicar os conceitos aplicados de aprendizado profundo (deep learning). Esse trabalho, inclusive, deu origem à criação do conjunto de dados MNIST. Essa é uma base de dígitos manuscritos, que é utilizada como conjunto de dados de referência com muita frequência nos trabalhos de visão computacional.

Atualmente, há muita pesquisa sendo feita nessa área, que combina o desenvolvimento de algoritmos de aprendizado de máquina, estruturas de dados e formas mais eficientes de processamento de hardware. Nesse cenário, as redes CNN são extremamente importantes e já vêm disponíveis em muitos pacotes computacionais. Na sequência, vamos explicar sobre um importante aspecto de ferramentas computacionais que vamos utilizar: framework.

Definição de framework

Você sabe o que é um framework?

Um framework é uma ferramenta de programação que já possui componentes prontos.

Assim, podemos desenvolver soluções muito mais rapidamente do que se tivéssemos que implementar todos os detalhes de um algoritmo. Em termos práticos, seria inviável ter que construir todos os detalhes de um projeto, pois consumiria muito tempo de desenvolvimento e de testes, além de ser necessário ter à disposição pessoas com diferentes habilidades. Portanto, veremos como usar os frameworks é muito útil.

Um ponto importante que precisamos destacar é a diferença entre um framework e uma biblioteca.

Biblioteca

Uma biblioteca é formada por um conjunto de funcionalidades. Quando desenvolvemos um programa, fazemos chamadas para essas funções. É a ideia de usar código reutilizável.



Framework

No caso do framework, utilizamos trechos de código personalizados. Portanto, o framework pode incluir compiladores, bibliotecas e outros recursos que nos ajudam a desenvolver soluções.

Existem muitas vantagens de utilizarmos frameworks. Uma delas é a garantia de trabalharmos com funcionalidades que já foram testadas. Isso nos poupa tempo e aumenta a confiabilidade da solução que estamos construindo. Existem muitos tipos de frameworks disponíveis no mercado. Muitos deles são softwares de código aberto. Para escolhermos qual deles é o mais adequado, precisamos levar alguns critérios em consideração:

1 Funcionalidades

Esse é um critério fundamental. Um programa pode precisar fazer muitas tarefas. Portanto é melhor utilizar um framework que trate a maior quantidade delas. Caso contrário, teremos que usar muitas bibliotecas e, talvez, outros frameworks que vão aumentar a complexidade do nosso código.

2

Consistência

Essa característica se refere à padronização do trabalho. Ela é muito importante, em especial, quando trabalhamos com grandes equipes.

3

Documentação

Essa característica diz respeito à preocupação com informações, sobre a melhor forma de usar as funcionalidades do framework. Também estamos interessados em exemplos e atualizações.

4

Comunidade ativa

Essa característica trata do engajamento das pessoas na utilização do framework. Inclusive, essa é uma das grandes vantagens de trabalhar com Python, pois temos uma comunidade grande e que interage bastante. Isso é muito útil quando precisamos tratar dúvidas muito específicas.

No caso da visão computacional, é muito importante utilizarmos frameworks, pois precisamos de muitas funcionalidades relacionadas à matemática (principalmente de álgebra linear), estatística e técnicas de aprendizado de máquina. Felizmente, temos muitos bons frameworks e bibliotecas de código aberto para desenvolvermos nossas soluções. Na sequência, nós vamos estudar alguns desses frameworks e bibliotecas.

Importantes frameworks para visão computacional

Neste vídeo, vamos destacar alguns frameworks importantes para a visão computacional.



Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

Frameworks e bibliotecas para visão computacional

Nesse momento, seremos mais específicos sobre o uso dos frameworks e das bibliotecas, pois o nosso objetivo é aplicá-los para visão computacional. As funcionalidades que precisamos utilizar, normalmente, cobrem os seguintes aspectos:

- Visualização: destacar objetos que dificilmente seriam visíveis em uma imagem.
- Reconhecimento: detectar objetos específicos em uma imagem.
- Nitidez e restauração: aperfeiçoar a qualidade das imagens.
- Reconhecimento de padrões: ser capaz de realizar categorizações dos elementos de uma imagem. Essa característica permite distinguir e classificar itens.

- Recuperação: auxiliar a realizar buscas de imagens semelhantes em banco de dados.

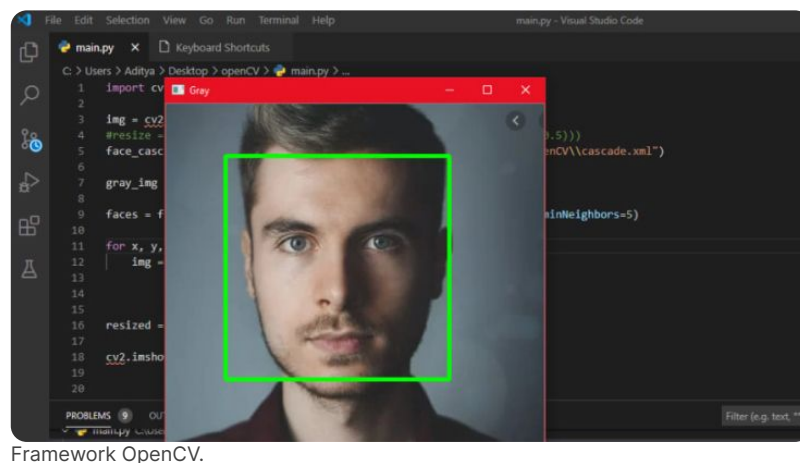
Todos esses aspectos que apresentamos são muito importantes na visão computacional e cada um deles precisa de diversas técnicas para alcançar a finalidade principal. Felizmente, temos à disposição frameworks e bibliotecas que facilitam nosso trabalho. Aqui, utilizaremos algumas dessas bibliotecas e frameworks com Python. Em especial, destacamos:

- OpenCV
- SciPy
- Pillow/PIL
- YOLO

Vamos dar uma olhada em cada uma delas!

OpenCV

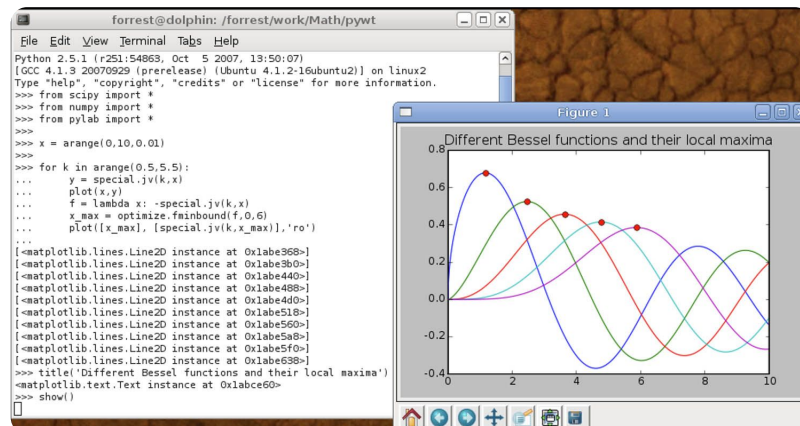
Essa é a biblioteca mais importante da atualidade voltada para aplicações de visão computacional. Ela não é limitada apenas para o Python. Foi desenvolvida e lançada pela Intel em 2000 e possui código aberto. Ela possui funcionalidades que podem ser usadas, por exemplo, para detecção facial e de objetos. Normalmente, quando a utilizamos, trabalhamos com outras bibliotecas como NumPy, SciPy e Matplotlib.



Framework OpenCV.

SciPy

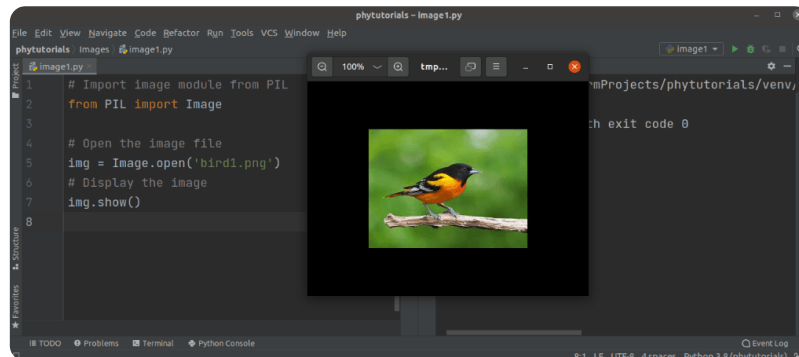
Também é uma biblioteca de código aberto e possui funcionalidades para realizar cálculos matemáticos e científicos com muita eficiência. No caso das aplicações de visão computacional, ela possui o submódulo `scipy.ndimage` que nos ajuda a realizar o processamento de imagens multidimensionais.



Biblioteca SciPy

Pillow/PIL

É uma biblioteca de código aberto voltada para tarefas de processamento de imagens. É um aperfeiçoamento da biblioteca PIL (Python Imaging Library). Essa biblioteca se destaca em relação às demais devido à facilidade de trabalhar com diferentes formatos de imagens.

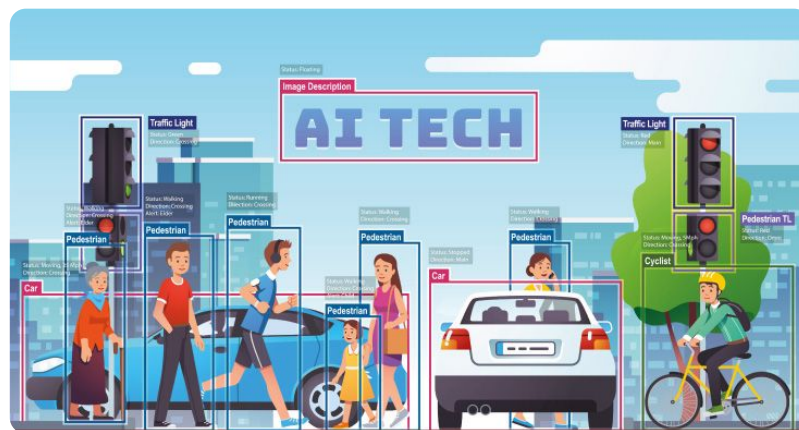


Framework Pillow/PIL.

YOLO

YOLO é uma sigla para “You Only Look Once”, que pode ser traduzido literalmente para “você só olha uma vez!”. Trata-se de um framework muito poderoso, voltado exclusivamente para visão computacional. Entre as diversas aplicações que ele permite construir está a detecção de objetos em tempo real. Isso só é possível porque os algoritmos que ele utiliza são extremamente eficientes. É um dos exemplos do sucesso dos algoritmos e arquiteturas de aprendizado de máquina para resolver problemas reais. Ele aplica uma rede neural CNN para classificar os objetos.

Na sequência, vamos estudar mais uma biblioteca e um poderoso framework que pode ser usado para além das aplicações de visão computacional. Ambos são muito importantes no contexto que estamos estudando.



Sistema de detecção de objetos de visão e vigilância por computador.

Biblioteca Scikit-Image e framework Tensorflow

Confira este vídeo sobre as principais bibliotecas para processamento de imagens e aprendizado de máquina.



Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

O poder da comunidade na construção de frameworks

É muito comum vermos um destaque muito grande para as aplicações de visão computacional por meio do uso de algoritmos de aprendizado de máquina e do avanço tecnológico. No entanto há algo extremamente importante que viabilizou esse avanço: o engajamento da comunidade.

Quando falamos de comunidade, estamos nos referindo a diversas pessoas, como pesquisadores de instituições públicas e privadas e programadores profissionais e amadores. Em especial, essa interação da comunidade foi impulsionada pela popularização da linguagem de programação Python que possibilitou a incorporação de muitas bibliotecas e frameworks. Dessa forma, ficou muito mais rápido testar os algoritmos e aperfeiçoá-los. Outro fator importante nesse processo de engajamento foi a popularização de repositórios de código públicos, como o GitHub.



Exemplo

Esse engajamento da comunidade voltado para visão computacional foi demonstrado no desenvolvimento da biblioteca Scikit-image. Ainda temos o caso do framework Tensorflow que possui muitas funcionalidades voltadas para visão computacional.

No entanto, o escopo de aplicação dele é muito mais amplo. Inclusive, o Tensorflow é, atualmente, o framework mais importante que existe voltado para aplicações de aprendizado de máquina. A seguir, vamos entender um pouco mais sobre o Scikit-image e o Tensorflow.

Scikit-image

É uma biblioteca do Python voltada para processamento de imagens. Ela utiliza matrizes no padrão do pacote NumPy para manipular dados. Entre seus recursos de destaque estão o uso de implementações eficientes para processamento de imagens e técnicas de visão computacional, funções fáceis de trabalhar, construído sobre NumPy, SciPy e matplotlib e possui código aberto, comercialmente utilizável sob a licença BSD.

TensorFlow

É o framework mais importante da atualidade voltado para aplicações de aprendizado de máquina e, mais especificamente, para aprendizado profundo (deep learning). Apesar de poder ser utilizado para aplicações de visão computacional, também é utilizado para diversos outros tipos de aplicações, como processamento de linguagem natural e processamento e reconhecimento de voz. Semelhante ao que ocorre com o OpenCV, o Tensorflow também pode ser utilizado por diversas linguagens de programação como Python, C, C++, Java e JavaScript. Ele possui uma documentação muito detalhada com diversos exemplos práticos. Sem dúvidas, é um conhecimento obrigatório para os que desejam se aprofundar nas aplicações de aprendizado de máquina.

Por enquanto, vimos uma parte mais teórica de visão computacional. Mais adiante, vamos aplicar esses conceitos. Para isso, vamos utilizar a linguagem de programação Python e o ambiente de desenvolvimento Google Colab. Antes de partir para a prática, reveja os conceitos e faça os exercícios para consolidar o seu conhecimento.

Verificando o aprendizado

Questão 1

Os algoritmos de visão computacional são muito complexos, portanto é muito importante utilizar ferramentas, como frameworks, que facilitem o processo de desenvolvimento. Nesse sentido, selecione a opção correta a respeito dos aspectos essenciais de um framework voltado para visão computacional:

A

Um framework deve ter apenas exemplos de modo que os usuários possam reproduzi-los.

B

Sempre devemos optar por frameworks com licença paga devido à confiabilidade que oferecem.

C

Frameworks e bibliotecas são equivalentes por possuírem funcionalidades que podem ser invocadas dentro do código principal.

D

É essencial que, além da estrutura das funções necessárias para operações de visão computacional, também haja documentação com exemplos.

E

Um dos maiores riscos de utilizar frameworks é o de perder a padronização no desenvolvimento de um projeto que envolve vários profissionais.



A alternativa D está correta.

Os frameworks são instrumentos essenciais para desenvolver aplicações de visão computacional eficientes. Eles oferecem recursos que padronizam o processo de programação. Além disso, é essencial utilizar como critério de escolha de um framework a existência de uma boa documentação com a disponibilidade de diversos exemplos práticos.

Questão 2

Atualmente, temos à nossa disposição muitos recursos sofisticados que facilitam o processo de desenvolvimento de aplicações complexas. Um bom exemplo disso são os frameworks e bibliotecas voltadas para visão computacional. Nesse sentido, selecione a opção correta a respeito dos frameworks e das bibliotecas específicas para aplicações de visão computacional:

A

São focadas na visualização dos elementos de uma imagem.

B

Todas tentam reproduzir o comportamento humano ao detectar algo diferente em uma imagem.

C

Ambos os recursos possuem muitas limitações práticas condicionando, assim, o uso deles apenas para casos muito específicos.

D

Os frameworks oferecem funcionalidades para as bibliotecas, de modo que possam ser utilizadas nos códigos das aplicações de visão computacional.

E

Entre as aplicações de frameworks e bibliotecas está a melhoria da qualidade das imagens por meio de um processo padronizado.



A alternativa E está correta.

Os frameworks e as bibliotecas são poderosos recursos computacionais que viabilizam a padronização e reutilização de código. Existem diversas funcionalidades que eles podem oferecer. Entre elas estão aumentar a qualidade de uma imagem e reconhecer padrões.

Aspectos de reconhecimento e detecção de imagens

Neste vídeo, destacaremos alguns pontos interessantes de aspectos de reconhecimento e detecção de imagens.



Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

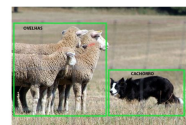
Definição de reconhecimento de imagens

O reconhecimento de imagens consiste no processo de identificação de imagens e, na sequência, na categorização delas em uma classe que já foi predefinida. Por isso mesmo ela também é conhecida como classificação de imagens. As aplicações que utilizam reconhecimento de imagens são capazes de definir o que está representado na imagem em questão e conseguem distinguir os objetos que estão dentro dela.

Esse trabalho, inclusive, é o foco da visão computacional, ou seja, habilitar sistemas artificiais com a capacidade de reconhecer objetos. Podemos utilizar esse tipo de aplicação com diferentes finalidades, veja a seguir.

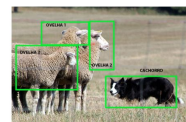
Classificação de imagens com localização

Nesse caso, a imagem é agrupada em classes mediante uma caixa delimitadora em torno dos objetos em questão para destacar a localização deles.



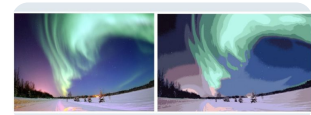
Detecção de objetos

Aqui, temos uma evolução sobre a classificação de imagens, pois os objetos são tratados de forma distintas. Também utilizamos caixas delimitadoras para representá-los.



Segmentação semântica

Existem ainda outras formas mais sofisticadas de reconhecimento de imagens utilizadas especialmente para robótica, como a segmentação semântica. Aqui temos um exemplo que demonstra a imagem original (à esquerda) e a imagem segmentada (à direita).



Definição de detecção de imagens

A detecção de imagens é o conjunto de técnicas que proporcionam a capacidade de uma tecnologia realizar o processamento de uma imagem e encontrar objetos específicos nela, ou seja, de detectar objetos. Há uma pequena confusão entre detecção e classificação de imagens.

Classificação

Técnica de categorizar um item em uma classe previamente definida. Por exemplo, existem pessoas dentro de uma imagem.

Detecção

Aqui devemos ser mais precisos sobre qual a quantidade de pessoas dentro da imagem e, em alguns casos, quem são essas pessoas. As técnicas de detecção podem ser utilizadas em diversos contextos, como localizar indivíduos desaparecidos e distinguir entre itens verdadeiros e falsificados.

Atualmente, a detecção de imagens ganhou uma importância muito maior devido aos avanços da tecnologia de big data, que é capaz de armazenar e gerar um volume gigantesco de dados — na ordem de petabytes. Dessa forma, podemos utilizar essas técnicas para localizar pessoas, perceber situações perigosas e agir preventivamente.

Abordaremos a seguir alguns aspectos sobre as principais tendências e tecnologias de reconhecimento de imagens.

Tendências na visão computacional e reconhecimento de imagem

A visão computacional já teve muitos avanços, mas ainda existem muitas melhorias que precisam ser alcançadas. No vídeo a seguir, falaremos sobre as principais tendências da área.



Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

Principais tendências

A visão computacional já teve muitos avanços, mas ainda existem muitas melhorias que precisam ser alcançadas. Por se tratar de uma área muito dinâmica, há muita pesquisa sendo desenvolvida nessa área. Isso, inclusive, é um grande incentivo para nos envolvermos mais com esse assunto, pois também temos oportunidades de produzirmos contribuições significativas. Atualmente, as principais tendências dessa área são:

- Extração de recursos.
- Histograma de gradientes orientados.
- Reconhecimento aéreo de alta altitude.
- Rede neural convolucional (CNN).

Vamos conferir brevemente cada uma dessas tendências.

Extração de recursos

Tem como objetivo utilizar a quantidade mínima de características (redução do número de recursos) que possa descrever um grande conjunto de dados. Em especial, quando analisamos imagens de objetos complexos, precisamos nos preocupar com a utilização de técnicas que sejam eficientes para detectar e reconhecer os elementos essenciais de uma imagem, de modo a distingui-la de outros elementos da imagem.

Histograma de gradientes orientados

Histograma é um recurso muito utilizado em estatística para compreendermos a distribuição dos dados. De forma semelhante, nós também utilizamos histogramas em aplicações de processamento de imagens para a detecção de objetos presentes em uma imagem ou vídeo. Nesse caso, utilizamos a nomenclatura histograma de gradientes orientados.

Reconhecimento aéreo de alta altitude

Esse tipo de técnica é especialmente útil para imagens áreas e de satélites. Seu objetivo é reconhecer objetos e situações que caracterizem anomalias, como desmatamentos, por exemplo. Entre as técnicas utilizadas aqui estão a detecção de bordas, linhas e retângulos.

Rede neural convolucional (CNN)

Já comentamos sobre as redes neurais convolucionais (CNN) anteriormente. É uma classe de redes neurais de aprendizado profundo (deep learning). Atualmente, é o modelo computacional mais importante aplicado para visão computacional. Basicamente, ela combina duas categorias de camadas: uma delas é responsável pela extração de características e a outra é responsável pela classificação.

Há muitas outras tendências sobre visão computacional, mas as que vimos já nos dá uma boa ideia de como essa área é dinâmica e cheia de possibilidades. A seguir, vamos desenvolver dois exemplos práticos para demonstrá-las.



Dica

Todos os exemplos foram desenvolvidos no Google Colab. É uma excelente ferramenta para trabalhar e muito simples de usar. Tente executar os passos que você verá para adquirir mais confiança e avançar nos seus conhecimentos sobre visão computacional.

Exemplo prático

Neste vídeo, utilizaremos o Google Colab na execução de um exemplo prático de processamento de imagem.



Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

Imagem colorida x imagem em tons de cinza

O objetivo desse exemplo é coletar uma imagem colorida online. Em seguida, o programa vai transformar a imagem colorida para tons de cinza e exibir, lado a lado, a imagem original e a modificada. Para isso, utilizamos a linguagem Python e, em especial, exploramos a biblioteca cv2. cv2 é o nome de importação do módulo para opencv-python. O exemplo é dividido nas seguintes partes:

- Importar pacotes.
- Importar imagem.
- Exibir a imagem processada.

Importar pacotes

Os pacotes mais importantes que precisamos utilizar são o numpy, cv2, skimage, google.colab.patches e matplotlib.pyplot. Em especial, o cv2 é o mais importante de todos para uma aplicação de visão de computacional. A seguir, apresentamos o código:

```
python

import numpy as np
import cv2
from skimage import io
from google.colab.patches import cv2_imshow
import matplotlib.pyplot as plt
```

Importar imagem

Depois de importar os pacotes, precisamos carregar a imagem. Basicamente, escolhemos um endereço com uma imagem com extensão png. Em seguida, carregamos a imagem original e aplicamos uma transformação para tons de cinza, conforme podemos ver no código a seguir:

```
python

origem = "https://upload.wikimedia.org/wikipedia/en/7/7d/Lenna_%28test_image%29.png"
img_original = io.imread(origem)
img_cinza = cv2.cvtColor(img_original, cv2.COLOR_RGB2GRAY)
```

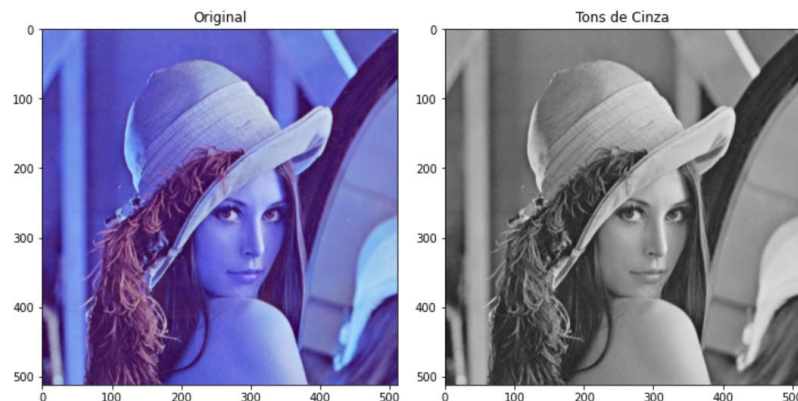
Exibir a imagem processada

Nosso próximo passo é exibir a imagem original (colorida) e a transformada (tons de cinza). Para isso, vamos usar as funcionalidades do pacote matplotlib e do cv2. Veja o código:

```
python

fig, ax = plt.subplots(1, 2,
figsize=(10, 6))
fig.tight_layout()
ax[0].imshow(cv2.cvtColor(img_original,
cv2.COLOR_BGR2RGB))
ax[0].set_title("Original")
ax[1].imshow(cv2.cvtColor(img_cinza,
cv2.COLOR_BGR2RGB))
ax[1].set_title("Tons de Cinza")
plt.show()
```

O resultado da execução do código é demonstrado nas imagens a seguir:



Esse exemplo é bem simples, mas apresenta elementos muito importantes para a manipulação de imagens e, posteriormente, para a detecção e destaque de características que, dificilmente, seriam percebidas sem o uso de ferramentas específicas.

Nesse momento, é fundamental que você tente executar o exemplo e realizar modificações para verificar o comportamento. Em especial, explore a biblioteca cv2. Na sequência, responda os exercícios e continue os seus estudos nessa fascinante área de visão computacional.

Verificando o aprendizado

Questão 1

A necessidade de obter resultados mais precisos e eficientes fez com que a visão computacional se dividisse em algumas subáreas. Essas especializações são importantes, pois permitem desenvolver técnicas mais eficientes, que se adequam bem a determinados contextos e que seriam inadequadas para outros cenários. Nesse sentido, selecione a opção correta a respeito sobre a utilização das técnicas de visão computacional:

A

Sempre garantem uma precisão que é impossível de ser alcançada pelo olho humano.

B

Utilizam algoritmos extremamente eficientes de modo que sempre produzem respostas rápidas.

C

Os algoritmos ficam limitados à detecção de objetos que já foram pré-cadastrados no sistema.

D

Não há como dar garantias sobre a qualidade dos resultados dos algoritmos de visão computacional, sendo necessário que haja verificação humana.

E

Pode ser utilizada para categorizar elementos em uma imagem em classes pré-cadastradas.



A alternativa E está correta.

Os algoritmos de visão computacional podem ser usados para diversas finalidades. Entre elas está a de classificação de itens cujas classes já foram previamente cadastradas no sistema. Isso não significa que o item em si foi cadastrado, mas sim a categoria a qual ele pertence.

Questão 2

Apesar dos grandes avanços das técnicas de visão computacional, ainda há muito trabalho a ser feito. Podemos ver isso claramente nas tendências de crescimento sobre essa área. Nesse sentido, selecione a opção correta a respeito das tendências sobre os avanços da visão computacional:

A

Todas as tendências sobre a área de visão computacional são igualmente importantes e promissoras.

B

O melhor caminho para se investir na evolução da visão computacional é no avanço dos equipamentos eletrônicos.

C

Investir no desenvolvimento dos modelos de redes neurais de aprendizado profundo tem se mostrado muito eficiente.

D

Ainda é muito cedo para decidir qual a melhor área para investir na evolução da visão computacional.

E

Devido às dificuldades em desenvolver algoritmos eficientes de visão computacional, o ideal é restringir a aplicação dos algoritmos para jogos de computador.



A alternativa C está correta.

A visão computacional tem sua história iniciada nos anos 1950. Portanto já é uma área bem consolidada com diversas técnicas bem testadas, e algumas delas têm se mostrado muito promissoras, como é o caso das redes neurais de convolução.

Casos de uso de reconhecimento de imagem

Aplicações práticas de visão computacional e reconhecimento de imagens

Neste vídeo, falaremos sobre as muitas possibilidades de aplicações práticas de visão computacional e reconhecimento de imagens e seus aspectos.



Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

Agora vamos nos aprofundar um pouco mais nessas aplicações, analisando, primeiramente, os seguintes casos práticos:

- Detecção de logotipo em análise de mídia social.
- Análise de imagens médicas.
- Reconhecimento de obras de arte e documentos originais.

Detecção de logotipo em análise de mídia social

Uma das principais formas usadas pelas empresas para divulgarem seus produtos e serviços é por meio da construção de uma logomarca.



Exemplo

Quando pensamos em um hambúrguer com batatas fritas e um refrigerante ou suco servido por um personagem fantasiado de palhaço, em que empresa pensamos? Ou ainda, na época do Natal, quando vemos o personagem do Papai Noel vestido de vermelho e preto, normalmente, bebendo um certo refrigerante, fazemos uma associação rápida entre marca e empresa.

Portanto é muito importante garantir padrões de qualidade, para que haja uma uniformidade no uso das logomarcas, como também é necessário monitorar outras empresas para verificar se estão tentando realizar cópias não autorizadas.

A visão computacional pode ser extremamente útil para monitorar as duas situações: padronização e tentativas de fraudes.

Essas duas aplicações são fáceis de serem percebidas e muito importantes, mas há situações ainda mais sofisticadas nas quais a visão computacional pode nos ajudar, em parceria com outras técnicas: associar quais adaptações de logotipos causam melhor receptividade no mercado consumidor. Dessa forma, a empresa passa a aumentar as suas vendas e os consumidores ficam satisfeitos com o que adquirem.

Análise de imagens médicas

Basicamente, os modelos computacionais passam por todo o processo de coleta de dados, extração de características e treinamento. Em seguida, são submetidos a diversos testes até que alcancem patamares de alta confiabilidade e possam ser utilizados para dar suporte aos profissionais de saúde para detectar e identificar anomalias.

Na área da saúde, existem diversas fontes de dados como radiografia, tomografia computadorizada, ultrassonografia e ressonância magnética.

Além disso, os profissionais dessa área sempre são muito pressionados para atender muitos pacientes e possuem pouco tempo para analisar tantos casos. Nesse sentido, a visão computacional pode direcionar os seus esforços para focar situações com maior probabilidade de ocorrer problemas e, assim, aumentar as chances de um diagnóstico mais preciso e de um tratamento mais eficaz. Cabe ainda destacarmos que o objetivo aqui não é o de substituir o profissional de saúde. Isso não faz nenhum sentido no mundo real. A ideia é utilizar os algoritmos de visão computacional como ferramentas de apoio à decisão.

Reconhecer obras de arte e documentos originais

Esse tipo de aplicação tem se tornado cada vez mais popular pois, com o avanço da tecnologia, houve um grande aumento das tentativas de fraudes. Além disso, essas tentativas estão cada vez mais sofisticadas e, em muitos casos, é praticamente impossível distinguir entre obras de arte e documentos originais dos casos de fraude. Portanto, a visão computacional já é utilizada para tratar situações desse tipo. Certamente, é mais uma das áreas que precisa avançar bastante, pois a tendência é que os fraudadores também continuem se sofisticando para aplicar golpes e prejudicar a sociedade.

Um ambiente favorável para muitas possibilidades de aplicações

Neste vídeo, escolhemos algumas aplicações para a visão computacional para relacionar a um ambiente favorável.



Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

É muito provável que você já tenha ouvido falar nos seguintes termos:

Big data

Tecnologia que gerencia (coleta, extrai, transforma e processa) grandes volumes de dados.

Internet das coisas

Aplicação da tecnologia da internet para interagir com dispositivos passivos (sensores) e ativos (acionadores).

Computação em nuvem

Tecnologia que faz todo o gerenciamento de sistemas e infraestrutura hospedados em servidores na internet.

Automação de processos

Utilização de técnicas com pouca interação humana capaz de sistematizar todo um processo de produção.

KPIs

Indicadores de desempenho-chave que são essenciais para medir a qualidade da execução dos processos.

Business intelligence

Técnicas de acompanhamento e suporte para tomada de decisão.

Ciência de dados

Combinação de conhecimentos que envolvem matemática, estatística, inteligência artificial, conhecimento prático dos chamados especialistas e uso de tecnologias voltados para obter o máximo de informações de qualidade que dão suporte à tomada de decisão.

Mineração de dados

Técnicas específicas para cruzar dados e elencar elementos fundamentais que serão usados posteriormente. É uma subárea da ciência de dados, mas com características que dão merecimento para um destaque especial.

Além de todos esses itens, temos ainda a popularização dos algoritmos de inteligência artificial, especialmente por causa dos algoritmos de aprendizado de máquina (machine learning) e dos algoritmos de aprendizado profundo (deep learning).

Essa combinação de tecnologias e conhecimentos nos dá possibilidades gigantescas de aplicações nas mais diferentes áreas. É bem verdade que o mercado ainda tem muitas limitações para absorver tantos assuntos e muitas empresas ainda operam como se estivessem na década de 1980 com tecnologias obsoletas. Ao mesmo tempo que isso é frustrante, também nos mostra como há muita coisa para ser feita e como temos grandes oportunidades de atuação como empreendedores. A seguir, vamos tratar sobre mais algumas aplicações relacionadas à visão computacional que podem servir para nós como fontes inspiradoras.

Manutenção preditiva

O objetivo é perceber a ocorrência de sinais que são fortes indicativos de que podem ocorrer problemas futuros nos equipamentos. Portanto, a visão computacional pode ser utilizada para monitorar alguns componentes e emitir alertas sempre que um cenário de risco se configurar. Dessa forma, os responsáveis pela manutenção poderão se planejar e evitar que ocorra um problema mais sério.

Inspeção de embalagens

Quando compramos uma caixa de remédios, por exemplo, esperamos que a quantidade de cápsulas esteja condizente com o que é informado no rótulo da embalagem. Mas como um ser humano pode realizar um controle desse tipo em grande volume? É simplesmente impossível. Essa é mais uma aplicação na qual a visão computacional pode auxiliar com altíssima precisão.

Leitura de códigos de barras

Hoje em dia, é uma das situações mais comuns pagarmos boletos bancários ou produtos nos supermercados com códigos de barras. Simplesmente passamos o produto por uma luz vermelha e temos acesso a todas as informações sobre esse produto. Outra tecnologia similar são os QR Codes. Essa é mais uma solução baseada em visão computacional.



Segurança patrimonial e pessoal

Essa solução é bem mais sofisticada do que as que tratamos até agora, pois reconhece padrões de comportamento. Além de envolver um complexo processo de treinamento do modelo computacional, precisamos de uma combinação de câmeras que forneçam dados em tempo real e, além disso, precisamos ter à disposição algoritmos e equipamentos que possam realizar processamentos muito eficientes para que os responsáveis pela segurança — física e patrimonial — possam analisar rapidamente a situação e atuar.

Reconhecimento de imagem em seguros

É uma aplicação muito útil tanto para o segurado como para a seguradora pois, no caso de um acidente, a vítima pode filmar o equipamento e enviar para que os prestadores de serviço possam atuar rapidamente. Apesar de existir essa tecnologia, infelizmente muitas empresas de seguro não a utilizam.



Dica

Que tal desenvolver um aplicativo nessa área e ajudar muitas pessoas, além de ter sucesso como empreendedor? Depois nos avise do seu sucesso!

Monitoramento de áreas ecológicas, agrícolas e de criação de animais

As imagens aéreas são úteis para monitorar a evolução de desmatamentos e queimadas. Além disso, também podem ser usadas para acompanhar as produções agrícolas e a criação e manejo de animais. Mais uma vez, a visão computacional nos dá a possibilidade de atuarmos com maior precisão e com a redução de custos.

Monitoramento do crescimento urbano

A população está em constante crescimento e isso gera enormes desafios para as prefeituras, pois são elas as responsáveis pelo desenvolvimento ordenado. Em especial, se isso não for feito com muito cuidado, podemos ver a ocorrência de diversos desastres, como infelizmente já ocorreram deslizamentos que provocaram enormes prejuízos materiais e, pior, a morte de muitas pessoas. Portanto, as aplicações de visão computacional são extremamente úteis para evitar essas situações desastrosas.

Vamos passar para o desenvolvimento de mais uma prática com o Python, utilizando um dos mais importantes frameworks para aprendizado de máquina: Tensorflow, sendo desenvolvido no Google Colab. Tente compreender o essencial do programa, pois, apesar de o framework facilitar o trabalho de desenvolvimento, ainda assim essas são aplicações bastante sofisticadas, que vão demandar esforço da sua parte, pesquisa na documentação oficial e nos fóruns na internet. Mas tenha certeza de que será um excelente investimento.

Uma aplicação de visão computacional com Tensorflow

Neste vídeo, utilizaremos o googlecolab em execução, em uma aplicação de visão computacional com Tensorflow.



Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

Informações básicas sobre a aplicação com o Tensorflow

Ao estudarmos sobre aprendizado de máquina, precisamos utilizar ferramentas que ofereçam muitos recursos para facilitar a nossa implementação. Caso contrário, teremos muitas dificuldades para realizar testes. Felizmente, temos à disposição frameworks como o Tensorflow. Inclusive, ele é o framework mais importante na atualidade voltado para aplicações de aprendizado de máquina. Além disso, não é restrito apenas à linguagem de programação Python, apesar de ser mais simples utilizá-lo com essa linguagem.

Aqui, vamos desenvolver uma aplicação bem interessante, cujo objetivo é reconhecer objetos de uso pessoal. Para isso, vamos usar uma base de dados chamada de “fashion_mnist” que já possui vários itens de roupas e calçados pré-cadastrados. A ideia é que a nossa aplicação será capaz de classificar corretamente outros itens que ela ainda não conhece explicitamente. Ou seja, ela vai utilizar a base de conhecimento e, a partir disso, fará a classificação. Nosso projeto está dividido nos seguintes itens:

- Importar bibliotecas.
- Carregar os dados.
- Definir o nome das classes.
- Explorar os dados.
- Exibir inicialmente os dados.
- Reescalar os dados para a escala 0-1 para redes neurais.
- Exibir algumas imagens.
- Construir a arquitetura da rede.
- Compilar o modelo.
- Treinar o modelo.
- Testar a precisão do modelo.

- Realizar previsões.

Importar bibliotecas

A primeira etapa é carregar o framework Tensorflow e as bibliotecas. A seguir, apresentamos o código:

```
python

import tensorflow as tf
from tensorflow import keras
import numpy as np
import matplotlib.pyplot as plt
```

Carregar os dados

O próximo passo é carregar os dados que vamos utilizar para treinar nosso modelo. No caso, vamos carregar a base "fashion_mnist" que já possui diversas imagens de itens pré-classificados. Nós precisamos separar os dados entre os que serão usados para treinar o modelo e os que serão utilizados para testar se o modelo está fazendo as classificações corretas. Veja o código a seguir:

```
python

fashion_mnist = keras.datasets.fashion_mnist
(treinamento_imagens, treinamento_rotulos), (teste_imagens, teste_rotulos) =
fashion_mnist.load_data()
```

Ao executarmos o código, recebemos a seguinte saída que nos informa de onde os dados estão sendo baixados e qual a quantidade de itens, conforme podemos ver na sequência:

```
Downloading data from
https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-labels-idx1-ubyte.gz
29515/29515 [=====] - 0s 0us/step

Downloading data from
https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-images-idx3-ubyte.gz
26421880/26421880 [=====] - 0s 0us/step

Downloading data from
https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-labels-idx1-ubyte.gz
5148/5148 [=====] - 0s 0us/step

Downloading data from
https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-images-idx3-ubyte.gz
4422102/4422102 [=====] - 0s 0us/step
Saída do código
```

Definir o nome das classes

Na sequência, precisamos escrever explicitamente os nomes das classes dos itens. Os nomes estão em inglês, pois foram armazenados dessa forma na base de dados. A seguir, apresentamos a lista de nomes que representam as classes utilizadas para rotular os itens:

```
python

nomes_classe = ['T-shirt/top', 'Trouser', 'Pullover', 'Dress', 'Coat', 'Sandal', 'Shirt', 'Sneaker', 'Bag', 'Ankle boot']
```

Explorar os dados

Aqui, passamos a entender um pouco melhor sobre os dados que estamos trabalhando. Basicamente, exibimos informações sobre os totais das imagens de treinamento e de teste, além das dimensões delas, conforme podemos ver no trecho de código a seguir:

```
python

print(f'(total de imagens, dimensão 1, dimensão 2)={treinamento_imagens.shape}')
print(f'rótulos={treinamento_rotulos} - São rotulados nas classes 0, 1, 2, 3, ...,9')
print(f'(total de imagens para teste, dimensão 1, dimensão 2)={teste_imagens.shape}')
```

Ao executarmos o código, vamos obter a seguinte saída:

```
(total de imagens, dimensão 1, dimensão 2)=(60000, 28, 28)
rótulos=[9 0 0 ... 3 0 5] - São rotulados nas classes 0, 1, 2, 3, ...,9
(total de imagens para teste, dimensão 1, dimensão 2)=(10000, 28, 28)
```

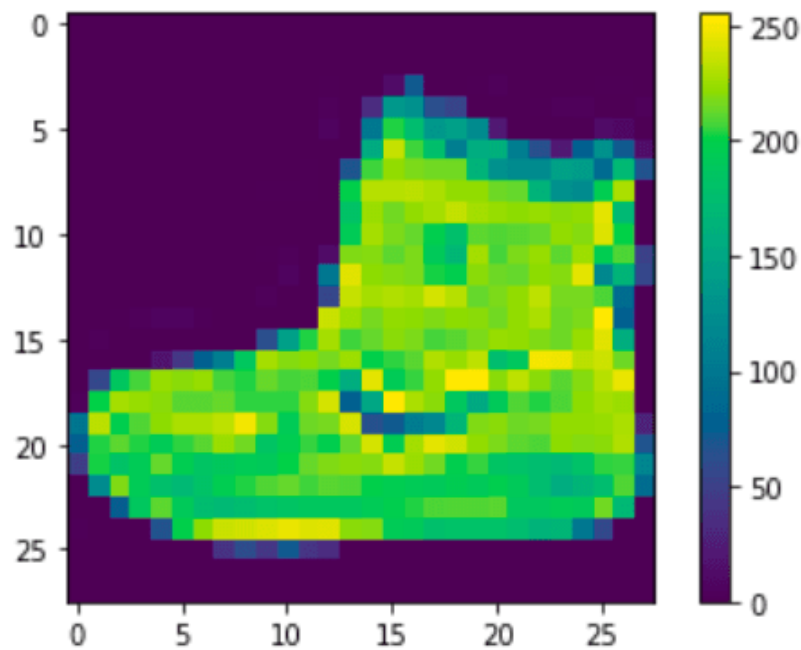
Exibir inicialmente os dados

Agora, vamos exibir uma das imagens da base de treinamento apenas para verificarmos se os dados vieram corretamente. Para isso, basta executarmos o seguinte código:

```
python

plt.figure()
plt.imshow(treinamento_imagens[0])
plt.colorbar()
plt.grid(False)
plt.show()
print('Explicação: exibindo a primeira imagem da base. Os dados numéricos se referem aos Pixels de 0 a 255.')
```

O resultado da execução do código é a imagem a seguir:



Exemplo de uma imagem da base de dados. Exibindo a primeira imagem da base. Os dados numéricos se referem aos pixels de 0 a 255.

Reescalar os dados para a escala 0-1 para redes neurais

Nosso próximo passo é realizar um ajuste nos dados, é o que chamamos de reescala. Basicamente, precisamos transformar os dados para o intervalo de 0 a 1, pois a rede neural precisa trabalhar com valores nessa faixa. Esse processo é bem simples, basta dividir os dados originais por 255, conforme podemos ver no trecho de código:

```
python

treinamento_imagens = treinamento_imagens / 255.0
teste_imagens = teste_imagens / 255.0
```

Exibir algumas imagens

Ainda com o objetivo de explorar os dados, vamos exibir algumas imagens. Nesse caso, precisamos executar o seguinte código:

```
python

plt.figure(figsize=(13,13))
for i in range(20):
    plt.subplot(5,5,i+1)
    plt.xticks([])
    plt.yticks([])
    plt.grid(False)
    plt.imshow(treinamento_imagens[i], cmap=plt.cm.binary)
    plt.xlabel(nomes_classes[treinamento_rotulos[i]])
plt.show()
```

A seguir, podemos ver o resultado da execução do código:



Exemplos de imagens da base do fashion_mnist.

Construir a arquitetura da rede

Nesse momento, vamos construir a arquitetura do nosso modelo de rede neural. Vamos utilizar três camadas: uma para sequenciar os dados e duas outras camadas densas para combinar as ligações entre os neurônios da rede, com o objetivo de melhorar o processo de aprendizado. Na prática, simplesmente precisamos executar o código a seguir:

```
python
modelo = keras.Sequential([
    keras.layers.Flatten(input_shape=(28, 28)),
    keras.layers.Dense(128, activation=tf.nn.relu),
    keras.layers.Dense(10, activation=tf.nn.softmax)
])
```

Compilar o modelo

Depois de construir a arquitetura da rede neural, precisamos compilá-la. Para isso, executamos o seguinte código:

```
python
modelo.compile(optimizer='adam',
    loss='sparse_categorical_crossentropy',
    metrics=['accuracy'])
```

Treinar o modelo

Agora, chegou a hora de treinarmos a rede. Para isso, basta usarmos a função “fit” do nosso modelo computacional e fornecer os parâmetros de treinamento, conforme podemos ver no código:


```
python

modelo.fit(treinamento_imagens,
           treinamento_rotulos,
           epochs=50)
```

Abaixo, apresentamos um trecho da execução do processo de treinamento. Os dados exibidos se referem ao processo de aprendizado do modelo:

```
1875/1875 [=====] - 6s 3ms/step - loss: 0.4463 - accuracy: 0.8482
Epoch 46/50
1875/1875 [=====] - 6s 3ms/step - loss: 0.4338 - accuracy: 0.8514
Epoch 47/50
1875/1875 [=====] - 5s 3ms/step - loss: 0.4426 - accuracy: 0.8443
Epoch 48/50
1875/1875 [=====] - 5s 3ms/step - loss: 0.4319 - accuracy: 0.8521
Epoch 49/50
1875/1875 [=====] - 5s 3ms/step - loss: 0.4313 - accuracy: 0.8524
Epoch 50/50
1875/1875 [=====] - 7s 3ms/step - loss: 0.4234 - accuracy: 0.8539
Trecho de saída do processo de treinamento.
```

Testar a precisão do modelo

Agora que o modelo já foi treinado, podemos conferir a qualidade dos testes. No caso, a precisão dos testes ficou em torno de 0.1422. Esse valor é melhor quando ele estiver mais próximo de 1, mas, no nosso caso, fizemos apenas para testar. A seguir, apresentamos o trecho de código:

```
python

perda_teste, precisao_teste = modelo.evaluate(teste_imagens, teste_rotulos)
print('Precisão dos Testes:', precisao_teste)

313/313 [=====] - 1s 2ms/step - loss: 2.8877 - accuracy: 0.1422
Precisão dos Testes: 0.14219999313354492
```

Realizar previsões

Por fim, podemos testar o nosso modelo. Primeiro, precisamos chamar o método “predict”, conforme o seguinte trecho de código:

```
python

predicoes = modelo.predict(teste_imagens)
```

A saída é dada por:

```
313/313 [=====] - 1s 2ms/step
```

Agora, vamos verificar a qualidade de uma das classificações do modelo. Vamos escolher a primeira delas, ou seja, vamos simular o reconhecimento da primeira imagem do array. Basta executarmos o código a seguir:

```
python

predicoes[0]
```

A saída é o vetor com as seguintes probabilidades de classificação.

```
array([0.34265953, 0.00656203, 0.12865795, 0.097398 , 0.01141852, 0.02494858, 0.33346367, 0.01317012,
0.04000533, 0.00171633], dtype=float32)
```

No caso, precisamos selecionar a probabilidade mais alta. Para isso, executamos o código:

```
python

melhor_classificacao=np.argmax(predicoes[0])
print(f'Posição do melhor resultado da primeira imagem de teste: {melhor_classificacao}')
```

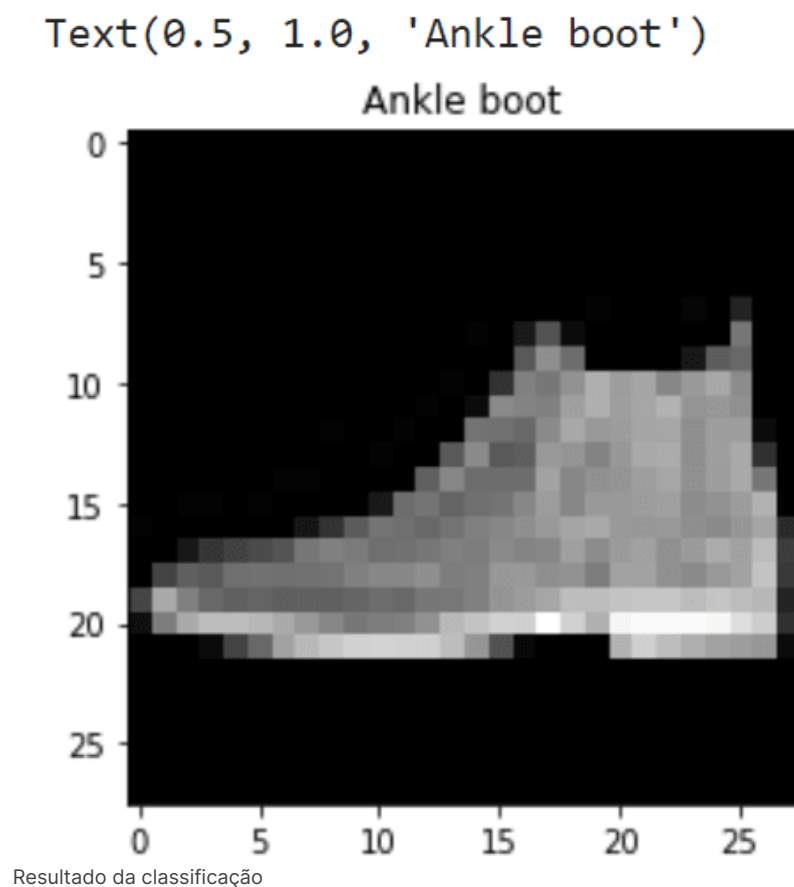
Posição do melhor resultado da primeira imagem de teste: 0

O resultado é, de fato, o que esperávamos. A seguir, apresentamos o código que exibe o resultado da classificação:

```
python

plt.figure(figsize=(14,4))
plt.imshow(np.reshape(teste_imagens[melhor_classificacao], (28,28)), cmap=plt.cm.gray)
plt.title(nomes_classes[teste_rotulos[melhor_classificacao]])
```

A imagem corresponde a uma bota de tornozelo (ankle boot), conforme podemos ver a seguir:



O ideal é que você tente reproduzir esse exemplo passo a passo. Depois, faça modificações e analise como o algoritmo se comporta.



Dica

A vantagem de utilizar o Tensorflow para desenvolver aplicações de visão computacional é que você terá muita facilidade de encontrar muita documentação e exemplos disponíveis na internet. É essencial praticar. E, mesmo que você cometa alguns erros pelo caminho, eles também fazem parte do aprendizado e serão muito úteis para consolidar seus conhecimentos. Bons estudos!

Verificando o aprendizado

Questão 1

A vida moderna exige que façamos muitas tarefas ao mesmo tempo, mas ainda não inventaram (até o dado momento) uma máquina para dilatar o tempo. Por isso mesmo, precisamos cada vez mais de ferramentas e métodos de planejamento que nos ajudem a ser mais eficientes. Uma situação muito interessante sobre eficiência ocorre na aplicação de técnicas de visão computacional na área médica. Nesse sentido, selecione a opção correta a respeito desse tipo de aplicação:

A

Uma das vantagens da visão computacional é ser capaz de substituir completamente profissionais da área de saúde.

B

É impossível confiar nos resultados produzidos por ferramentas de visão computacional no auxílio de diagnósticos médicos.

C

O maior limitante para a utilização plena de ferramentas de visão computacional na área médica é a falta de algoritmos confiáveis.

D

A visão computacional faz parte da inteligência artificial que, por sua vez, é uma das áreas da ciência da computação, o que a torna inadequada para a área de saúde.

E

A visão computacional pode ser um poderoso coadjuvante para auxiliar profissionais no direcionamento dos esforços de modo mais eficiente.



A alternativa E está correta.

A visão computacional pode ser extremamente útil para auxiliar profissionais de saúde a determinar o diagnóstico de pacientes por meio da análise de exames baseados em imagens. Já existem algoritmos

bastante sofisticados para essa finalidade. No entanto, é essencial entender que essas técnicas têm como objetivo apoiar o profissional de saúde, e não o substituir.

Questão 2

A utilização de ferramentas de visão computacional é cada vez mais comum no nosso dia a dia. Um exemplo disso ocorre com os códigos de barras e os QR codes, que são muito utilizados por supermercados. Nesse sentido, com base nas informações fornecidas na questão, selecione a opção correta:

A

Atualmente, os algoritmos estão muito sofisticados que são capazes de fazer leitura dos códigos com apenas uma parte deles.

B

Os algoritmos de leitura de códigos têm como objetivo primordial eliminar a necessidade de funcionários nos caixas de supermercados.

C

Para que a leitura dos códigos funcione corretamente é preciso garantir a integridade da imagem.

D

Apesar de ser uma excelente ideia, ainda não é possível confiar no uso dos algoritmos de leitura de códigos de barras.

E

A efetividade dos algoritmos de visão computacional voltados para leitura de códigos de barras depende da linguagem de programação escolhida para desenvolver os sistemas.



A alternativa C está correta.

A leitura de códigos de barras e de QR codes é uma das mais populares aplicações de visão computacional. Todos os dias, muitas pessoas fazem uso desse recurso, extremamente útil para agilizar o processo de controle de estoque e venda de produtos, além de outras aplicações. No entanto, é essencial compreender que os códigos precisam estar íntegros, caso contrário o algoritmo não poderá garantir a qualidade da classificação do produto.

Considerações finais

O nosso objetivo, ao longo desse trabalho, foi apresentar os conceitos essenciais de visão computacional e de reconhecimento de imagens. Além disso, também tivemos a oportunidade de explorar bibliotecas e frameworks que facilitam o trabalho de desenvolvermos aplicações práticas.

É muito interessante observarmos como diversos assuntos se relacionam — visão computacional, big data, computação em nuvem e internet das coisas, por exemplo — e nos dão a oportunidade de criarmos soluções que são muito úteis no dia a dia. Como ressaltamos ao longo desse trabalho, é fundamental praticar os exemplos apresentados e realizar modificações para aprofundar o conhecimento.

Podcast

Ouçá uma introdução à visão computacional e ao reconhecimento de imagem.



Conteúdo interativo

Acesse a versão digital para ouvir o áudio.

Explore +

Para aprofundar seus conhecimentos sobre o conteúdo abordado:

Acesse o site do **Scikit-image** e aprenda mais sobre as funções voltadas para visão computacional por meio da documentação oficial e de vários exemplos práticos.

Acesse o site oficial do **Tensorflow**. Lá, você encontrará exemplos práticos voltados para visão computacional e outras aplicações de aprendizado de máquina.

Referências

ASZEMI, N. M.; DOMINIC, P. D. D. **Hyperparameter optimization in convolutional neural network using genetic algorithms**. Int. J. Adv. Comput. Sci. Appl., v. 10, n. 6, p. 269–278, 2019.

FUKUSHIMA, K. **Neocognitron**: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biol. Cybernetics, v. 36, p. 193–202, 1980.

HUBEL, D.; WIESEL, T. **Receptive Fields and Functional Architecture of Monkey Striate Cortex**. J. Physiol., v. 195, p. 215–243, 1968.

IMAGENET. Consultado na internet em: 14 dez 2022.

LECUN, Y. *et al.* **Gradient-based learning applied to document recognition**. Proceedings of the IEEE, v. 86, p. 2278–2324, 1998.

SIMONYAN, K.; ZISSERMAN, A. **Very deep convolutional networks for large-scale image recognition**. ICLR, 2015.