



APLICATIVO PARA RECONHECIMENTO DE PONTOS TURÍSTICOS

GABRIEL GALLI^{1,2}, RAPHAEL BORGES^{2,3}, JOSÉ BINS^{2,4}

1 Introdução/Justificativa

A área de visão computacional (CV) se propõe a desempenhar tarefas do sistema visual humano (SVH) num contexto computacional, automatizado, como detecção, rastreamento e classificação de objetos. Além disto, a CV consegue realizar algumas tarefas que o SVH não é capaz de executar, como a captura de imagens médicas (radiografia, ressonância magnética, etc.). Assim, algumas vezes, um sistema computacional consegue ser melhor que o humano, mas, em muitas tarefas, o SVH ainda não foi superado (1, p. 3). Uma destas tarefas é a de classificação, ou seja, determinar se um dado objeto faz parte de uma classe conhecida.

Assim como no SVH o cérebro recebe informações visuais vindas dos olhos, um sistema de CV também recebe informações de alguma fonte, chamada entrada, como uma câmera, um arquivo local, alguma transferência em tempo real, algum sensor especializado, etc. Sobre a imagem são realizadas várias operações para extrair características relevantes e assim tentar determinar o que está retratado nela (1, p. 5). Porém, esta tarefa é muito difícil pois estamos tentando recuperar informações desconhecidas e ambíguas e por isso precisamos utilizar métodos que nos ajudem a remover possíveis respostas incorretas (1, p. 3).

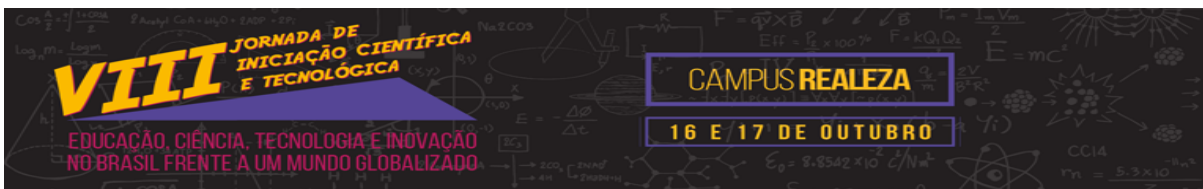
Existem várias técnicas para filtrar as possíveis respostas ao problema. Uma delas é chamada de *clustering* e consiste em agrupar informações segundo algum critério de similaridade. Por exemplo, no contexto de fotografias capturadas com *smartphones*, podemos utilizar o recurso de geolocalização do dispositivo, através de *geotagging* (veja (2)), para agrupar imagens pela proximidade geográfica. Entretanto, afirma (3), esta geolocalização enfrenta problemas de precisão e disponibilidade e (4) completa dizendo que ainda não alcançamos uma qualidade suficiente com os grupos (*clusters*) construídos a partir disto. Outra técnica consiste em detectar e extrair algum rótulo (*label*) que forneça dicas sobre do que se trata a imagem. Estes rótulos podem ser encontrados no nome do arquivo, em

1 Estudante da Universidade Federal da Fronteira Sul, *campus* Chapecó, **Bolsista** , contato: ggabriel96@hotmail.com.

2 Grupo de Pesquisa de Inovação e Desenvolvimento Tecnológico

3 Estudante da Universidade Federal da Fronteira Sul, *campus* Chapecó, contato: rah-borges@hotmail.com.

4 Professor Doutor da Universidade Federal da Fronteira Sul, *campus* Chapecó, **Orientador**, contato: jose.binsfilho@uffs.edu.br.



comentários ou até contidos na própria figura. Porém, técnicas como esta demandam um grande volume de exemplos (imagens com rótulos), o que é caro e trabalhoso para construir.

Tendo isso em mente, a ideia deste trabalho é utilizar várias destas técnicas em conjunto de modo a adicionar robustez às suas falhas individuais. Existem propostas similares, mas que dependem fortemente de técnicas avançadas de busca, indexação e *machine learning*, além do grande volume de dados citado anteriormente. Neste caso, a comunidade de usuários é responsável pela construção do banco de dados.

2 Objetivos

O objetivo geral é construir um aplicativo Android que funcione como uma rede social fotográfica com a proposta de identificar o ponto turístico retratado nas fotografias. Os objetivos específicos são: estudar o estado da arte da CV aplicada ao reconhecimento de imagens; estudar, comparar e aplicar algoritmos de detecção e descrição de características de imagens; utilizar *clustering* para filtrar imagens candidatas à comparação; estudar a plataforma Android; analisar e utilizar técnicas de comunicação cliente-servidor.

3 Material e Métodos/Metodologia

A identificação de um ponto turístico baseia-se em fotografias prévias do local. Como um aplicativo Android, através do Google Play Services temos acesso à localização do dispositivo para aplicarmos *geotagging* e construímos *clusters* para realizar um filtro sobre a localização da fotografia. Também dispomos das medições realizadas pelos vários sensores disponíveis, como acelerômetro, giroscópio e magnetômetro, que nos proporcionam recuperar a orientação do dispositivo em relação ao norte magnético da Terra (ou seja, descobrir em que direção ele estava apontado quando capturou a imagem). Esta direção nos possibilita realizar um *clustering* e então um filtro adicional após o da geolocalização. Os dois filtros reduzem a quantidade de imagens com as quais é necessário comparar uma nova fotografia para tentarmos descobrir o ponto turístico.

A etapa de comparação é feita através da detecção, extração e *matching* de características relevantes em pares de imagens (uma nova fotografia e uma candidata do banco de dados). Cabe destacar aqui que realizamos mais um filtro (baseado na teoria estatística de remoção de *outliers* com média, desvio padrão e percentis e inspirado por (5)) para remover possíveis mapeamentos que são, na verdade, falsos positivos. São consideradas características relevantes aquelas que são invariantes à transformações aplicadas na imagem,



como escala, translação, rotação, iluminação, etc. Quanto mais invariante for uma característica, mais robusta ela será aos ruídos presentes na comparação, aumentando a eficácia do processo.

Existem vários algoritmos para detectar e descrever características de imagens. Dentre eles, estamos testando: SIFT, SURF, ORB, AKAZE e BRISK, da biblioteca OpenCV. Como não possuímos um banco de dados de pontos turísticos, estes testes são realizados sobre imagens que coletamos dos municípios de Chapecó (SC) e São Paulo (SP). Pela ausência dessa base de dados, aqui é importante enfatizar a parte social do aplicativo. No princípio, contaríamos exclusivamente com os rótulos atribuídos pelos usuários: dadas as respectivas configurações de permissões e privacidade, tanto o dono da foto quanto outros usuários podem indicar qual é o ponto turístico retratado na imagem. A partir destas indicações, os usuários podem votar nas sugestões existentes de rótulos e os mais votados são destacados. Tão logo quanto existirem fotografias em certo ponto turístico (rotulado), o sistema passará a sugerir rótulos para as novas imagens.

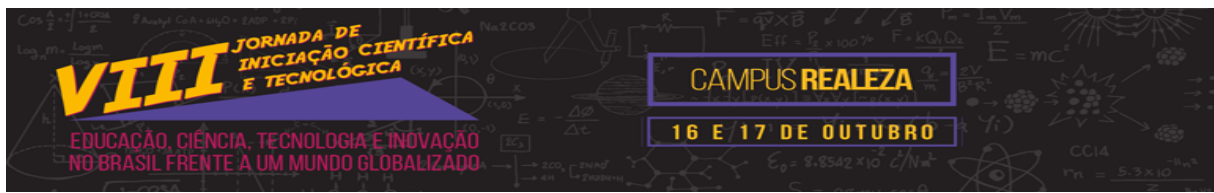
4 Resultados e Discussão

Um aplicativo protótipo foi criado. Nele há uma câmera embutida que salva a geolocalização e a orientação do dispositivo. Para o servidor, optamos por terceirizar o máximo de funções fora do escopo da CV. Desta forma, comparamos provedores de nuvem pública e o escolhido foi o Google Cloud Platform (GCP) por, dentre outras vantagens, oferecer profunda integração com o Android. Para cadastro e *login*, utilizamos o Firebase Authentication. O armazenamento de dados ocorre pelo Firebase Storage e GCP Datastore.

Em relação à comparação de imagens, das mais de 260 que coletamos, os últimos testes foram realizados sobre 36 pares, que representam objetos iguais e diferentes e com variação de ângulo de visão e escala, e os algoritmos de detecção e descrição de características ainda não ofereceram uma precisão satisfatória. Nos casos mais extremos, chegamos perto de 40% de falsos positivos. Por isso estamos trabalhando com a remoção de *outliers* (inspirado em (5) citado acima) e começamos a explorar algoritmos de classificação sobre as descrições (não sobre as imagens em si). Esta classificação, por sua vez, tem se mostrado promissora, com acurácia e precisão mais próximos de 90%.

5 Conclusão

Primeiramente, o projeto tem sido de grande aprendizado, principalmente nas áreas em



que nos propomos a estudar, e de ganho de experiência como pesquisadores. Buscar alternativas e justificar e embasar nossas escolhas são práticas que devem ser incentivadas e este projeto tem sido um exemplo muito claro disso. Nestes quesitos, as expectativas iniciais, com certeza, foram superadas. Além disso, as áreas de pesquisa que este trabalho envolve se mostram cada vez mais interessantes, até por serem de constante evolução e inovação.

Dentre as etapas realizadas, resultados obtidos até então e caminhos futuros, ressaltamos que o *clustering* não foi implementado, mas os dados estão disponíveis; a câmera interna do aplicativo precisa ser consideravelmente melhorada para atingir os níveis que estamos acostumados com este tipo de funcionalidade, mas é por si só complexa o suficiente para um projeto aparte; e, por fim, as interações de rede social também não foram implementadas, mas elas se tratam mais de um requisito trabalhoso e que requer tempo do que algo com potencial retorno científico como a proposta principal do projeto (embora estudar a magnitude do engajamento dos usuários nos permitiria melhor compreender a viabilidade do projeto como um serviço).

Referências

- 1 SZELISKI, Richard. **Computer Vision: Algorithms and Applications**. 1. ed. London, UK: Springer-Verlag, 2011. ISBN: 978-1-84882-935-0.
- 2 LUO, Jiebo et al. Geotagging in Multimedia and Computer Vision – a Survey. **Multimedia Tools Appl.**, Kluwer Academic Publishers, Hingham, MA, EUA, v. 51, n. 1, p. 187-211, jan. 2011. ISSN: 1380-7501. DOI: 10.1007/s11042-010-0623-y.
- 3 LIU, Heng et al. Accurate Sensing of Scene Geo-context via Mobile Visual Localization. **Multimedia Syst.**, Springer-Verlag New York, Inc., Secaucus, NJ, EUA, v. 21, n. 3, p. 255-265, jun. 2015. ISSN: 0942-4962. DOI: 10.1007/s00530-013-0344-y.
- 4 ZHU, Zhu; XU, Chang. Organizing photographs with geospatial and image semantics. **Multimedia Systems**, Springer Berlin Heidelberg, v. 23, n. 1, p. 53-61, fev. 2017. ISSN: 1432-1882. DOI: 10.1007/s00530-014-0426-5.
- 5 DOU, Jianfang; QIN, Qin; TU, Zimei. Robust image matching with cascaded outliers removal. **Pattern Recognition and Image Analysis**, v. 27, n. 3, p. 480–493, jul. 2017. ISSN: 1555-6212. DOI: 10.1134/S1054661817030099.

Palavras-chave: visão computacional; *geotagging*; *clustering*; rede social; Android.

Financiamento: CNPq.