



ARTIGOS COMPLETOS	2
RESUMOS DE PESQUISA	10
RELATOS DE EXPERIÊNCIA	23

ARTIGOS COMPLETOS

ANÁLISE SOBRE AS CARACTERÍSTICAS E O DESEMPENHO DO SISTEMA DE RECOMENDAÇÃO MREC..... 3

ANÁLISE SOBRE AS CARACTERÍSTICAS E O DESEMPENHO DO SISTEMA DE RECOMENDAÇÃO MREC

Matheus Amaral Da Silva e Kevin Levrone Rodrigues Machado Silva

Universidade Estadual de Maringá – UEM

E-mail: matheusrockway@gmail.com

RESUMO - Neste trabalho foi realizada uma análise das principais funcionalidades do sistema mrec na construção e avaliação de recomendações. Para avaliá-lo, foram reunidas todas as informações disponíveis referentes ao processo de recomendação. Com o intuito de avaliar o seu comportamento e desempenho, foi realizado um experimento com o conjunto de dados *MovieLens*, em que o sistema o divide em subconjuntos de treino e teste, utilizados para a criação/treinamento de modelos geradores de recomendações e para a avaliação dos modelos, respectivamente. O módulo de avaliação do mrec classifica as recomendações que forem geradas pelos modelos. A investigação das características e a análise de desempenho do mrec permitiram a obtenção de um veredito. Concluiu-se que é um *software* que cumpre sua proposta, possui funcionalidades que beneficiam os usuários, mas também, fatores negativos que influenciam desfavoravelmente na sua preferência e notoriedade.

Palavras-chave: sistemas; recomendação; python; mrec.

ANALYSING THE CHARACTERISTICS AND PERFORMANCE OF THE MREC RECOMMENDATION SYSTEM

ABSTRACT - In this work we analyzed the main features of the mrec system in the construction and evaluation of recommendations. To evaluate, we collected all the available information regarding the recommendation process. In order to evaluate behavior and performance, we performed an experiment with the *MovieLens* dataset, which the system divided it into training and test subsets, used for the creation/training of recommendation models and to test them, respectively. The system's evaluation module classifies the recommendations that are generated by the models. The characteristic investigation and performance analysis of mrec allowed us to obtain a verdict. We concluded that it is a software that fulfills its proposal, it has features that benefit users, but also negative factors that may adversely influence its preference and notoriety.

Keywords: systems; recommendation; python; mrec.

1. INTRODUÇÃO

Com o avanço do desenvolvimento e da demanda de novos *softwares* e sistemas *web*, surgiu-se a necessidade de implementar funcionalidades que evidenciassem tais serviços e, ao mesmo tempo, conquistasse a confiança do cliente. Desde então, uma das principais aplicações feitas com tal intuito envolve a integração de um serviço com um sistema de recomendação.

De acordo com (XIAO, BENBASAT, 2007), sistemas de recomendação são agentes de *software* que suscitam os interesses e preferências de um usuário e, com esses dados, fazem uma ou mais recomendações ao mesmo. Ao utilizar um sistema de recomendação robusto, há uma potencial melhora no suporte e na qualidade das decisões que os consumidores podem realizar enquanto buscam ou selecionam itens em algum serviço *online*.

Atualmente, encontram-se disponíveis quatro paradigmas de sistemas de recomendação personalizados que podem ser implementados nas mais diversas aplicações. Eles são classificados como Baseado em Filtragem Colaborativa, Baseado em Conteúdo, Baseado em Conhecimento e Híbridos. A maioria dos *frameworks* utilizados para o desenvolvimento de sistemas de recomendação já possuem uma implementação nativa de um ou mais dos paradigmas citados, cujos conceitos e funcionamentos são discutidos em detalhes no trabalho de Jannach, Zanker e Felfernig (2010).

O sistema de recomendação escolhido como objeto de estudo deste trabalho é o mrec (MREC, 2013). Com o desenvolvimento da pesquisa, foram reunidas todas as informações referentes ao seu funcionamento, criação e usabilidade, com o intuito de analisar suas funcionalidades e relevância na geração e avaliação de recomendações ao usuário. Para avaliar o seu desempenho e robustez, foi realizado um experimento com o conjunto de dados *MovieLens*. Ele foi escolhido como *dataset* para esse trabalho por se tratar do conjunto de dados utilizado como exemplo na documentação do mrec, além de ser um dos mais usados no meio acadêmico para realização de experimentos com sistemas de recomendação. Maiores detalhes sobre a composição do *MovieLens* são discutidos na Seção 3 e na sua página oficial (MOVIELENS, 2019).

O restante deste artigo está organizado como segue. Na Seção 2 é apresentado o sistema mrec. Os resultados da sua avaliação são apresentados na Seção 3. Por fim, na Seção 4, são apresentadas as considerações finais deste trabalho.

2. SOBRE O MREC

O mrec é um sistema de recomendação desenvolvido na linguagem de programação Python, e cuja elaboração foi realizada no gerenciador de referências Mendeley¹ para facilitar o desenvolvimento e a avaliação de recomendações.

O sistema possui funcionalidades que ainda lhe dão destaque entre outros sistemas semelhantes. Uma delas é a implementação nativa de ferramentas simples para a realização de avaliações consistentes e reproduzíveis de sistemas de recomendação. Outra funcionalidade consiste em utilitários para treinar modelos e fazer recomendações em paralelo via *IPython*. A presença de tais utilitários é incomum em *softwares* similares ao mrec, assim como materiais com exemplos de combinação de uso do *IPython* com outras bibliotecas científicas da linguagem Python.

Para os processos de geração e avaliação de recomendações, o mrec possui algoritmos que seguem o paradigma Híbrido ou o paradigma de Filtragem Colaborativa. Os pertencentes ao primeiro paradigma suprem limitações consequentes do uso individual de entradas e componentes de mecanismos de outros paradigmas nessas tarefas, fazendo com que combine algumas dessas entradas e componentes. Já os que se enquadram no segundo utilizam como entrada uma matriz de avaliações usuário-item e, para recomendar um determinado item a um determinado usuário, consideram as avaliações de todos os usuários que tiverem preferências semelhantes a ele em relação a outros itens que já foram avaliados anteriormente por esse mesmo indivíduo.

2.1 ALGORITMOS NATIVOS

Nesta seção são apresentados os algoritmos de recomendação do sistema mrec.

Popularity: Um algoritmo que gera as mesmas recomendações para todos os usuários, e que, de acordo com a documentação do mrec, pode ser útil para a etapa de avaliação (JANNACH, ZANKER, FELFERNIG, 2010).

K-Nearest Neighbours (KNN): O algoritmo é utilizado na geração de modelos tradicionais de similaridade de itens (DUDA, HART, STORK, 2012).

Sparse Linear Method (SLIM): É um algoritmo de recomendação *top-n* especializado na geração rápida de recomendações (NING, KARYPIS, 2011). Ele constrói uma matriz de coeficiente esparsa para os itens que estão no sistema ao utilizar apenas os perfis de compras ou avaliações de usuários (o que é possível por meio da resolução de um problema de otimização regularizada). Em seguida, é introduzida esparsidade na matriz de coeficiente, o que permite uma geração eficaz de recomendações. Ademais, o algoritmo SLIM também pode ser usado para efetuar recomendações *top-n* por meio de avaliações de itens.

Weighted Approximate-Rank Pairwise (WRMF): É modelo de fatoração de matriz de peso utilizado para conjunto de dados de *feedbacks* implícitos (HU, KOREN, VOLINSKY, 2008).

Weighted Approximate-Rank Pairwise loss (WARP loss): Algoritmo proposto originalmente para aplicação em conjuntos de dados de anotação de imagens, com o intuito de prever a anotação das mesmas. Consome menos memória e possui desempenho superior a métodos mais comuns que realizam

¹ <https://www.mendeley.com>

funções semelhantes (WESTON, BENGIO, USUNIER, 2010). O mrec possui a implementação de um modelo híbrido de matriz de fatorização que otimiza o WARP *loss*.

2.2 IPYTHON

Em 2001, o *IPython* surgiu como um projeto que almejava a criação de ferramentas para expandir as capacidades interativas da linguagem Python e, desde então, continua a ser desenvolvido como uma camada de base para novos ambientes interativos. A utilidade do *IPython* neste trabalho foi a de paralelização do treinamento de modelos, da geração e da avaliação de recomendações durante os experimentos com o mrec. Maiores informações sobre o projeto *IPython* e os seus recursos são apresentados na obra de Pérez e Granger (2007).

2.3 INSTALAÇÃO

A documentação oficial do mrec apresenta duas maneiras de instalá-lo em sistemas operacionais do tipo UNIX. A primeira necessita do gerenciador de pacotes pip, seguida da instalação do mrec pelo mesmo. Todavia, como foi removido dos repositórios do pip, tornou-se obrigatória a instalação por meio do seu código-fonte, que representa a forma alternativa de instalação.

Para instalar o mrec na distribuição Ubuntu 16.04 do Linux e no sistema operacional Os X v10.14.5, é preciso inicialmente instalar algumas de suas dependências:

- Python: 2.7
- Cython: 0.20
- Numpy: 1.11.0
- Scipy: 0.17.0
- scikit-learn: 0.20.3
- IPython: 4.0.0
- pyzmq: 18.0.1
- psutil: 3.4.2

Além disso, pode ser necessário realizar o *downgrade* da versão mais recente da dependência Cython (3.0) para o funcionamento correto do mrec.

2.4 MEDIDAS DE AVALIAÇÃO

De acordo com a documentação, para gerar e avaliar recomendações o mrec realiza, primeiramente, a divisão do conjunto de dados em subconjuntos de treino e de teste. Em seguida, é realizado o treinamento do(s) modelo(s) de recomendação com base no(s) subconjunto(s) de treino. Logo após, são geradas recomendações com base no(s) modelo(s), cujas avaliações serão baseadas no(s) subconjunto(s) de testes com base nas métricas *Mean Reciprocal Rank*, sendo esta uma medida estatística que avalia qualquer processo resultante na lista de probabilidade ordenada por precisão e pode ser descrita matematicamente da seguinte forma.

$$\text{MRR} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{\text{rank}_i}.$$

Outro método de avaliação é o *Precision@k*, onde *k* é o número de recomendações geradas pelo modelo. Um exemplo de uso, baseado na metodologia apresentada, é descrito a seguir na seção 3 onde é relatado o experimento e a sua avaliação.

2.5 USO DA MREC

Nesta seção é descrita a forma de execução do sistema e os comandos necessários para operá-la, é importante observar que este ainda não é o relato do experimento feito, trata-se de um *overview* das opções que o desenvolvedor tem ao escolher o mrec para utilizar em seus softwares.

Primeiramente, é necessário encontrar um conjunto de dados que possua as informações e formatações suportadas pelo mrec: tsv (valores separados por tabulação), csv, mm (*matrix market*) ou fsm (*fast sparse matrix*). Os valores normalmente podem ser organizados em uma tabela de avaliação item por usuário nos conjuntos de dados.

Em seguida, é preciso dividir o conjunto de dados em subconjuntos para treinar e testar o recomendador. O *mrec* seleciona alguns itens que o usuário avaliou positivamente e os move para o subconjunto de teste. Os itens restantes para cada usuário são reunidos no subconjunto de treino, utilizado para o treinamento do recomendador. O seguinte comando deve ser executado em um terminal do sistema operacional para realizar a divisão de um conjunto de dados (isto é, um *dataset*) com a formatação *tsv*:

```
mrec_prepare --dataset diretório_do_dataset dataset.tsv
```

Como apresentado na documentação do *mrec*, caso um utilizador do *mrec* sinta a necessidade de definir parâmetros adicionais para a divisão, é possível determinar o diretório de saída dos subconjuntos, quais itens serão divididos (com base na filtragem de um valor mínimo de avaliação pelo usuário), qual o tamanho desejado do subconjunto de testes, entre vários outros parâmetros. Antes de iniciar o treinamento do recomendador, é preciso inicializar a quantidade desejada de motores do *IPython* com o comando a seguir:

```
ipcluster start -n4 --daemonize
```

No caso do comando anterior, serão inicializados 4 motores. Caso não seja inserido um número na frente da opção *-n*, será iniciado um motor para cada núcleo existente no computador que estiver executando o experimento. O número de recomendadores que serão treinados em paralelo para cada divisão treino/teste é diretamente proporcional ao número de motores em execução. O treinamento de cada recomendador é realizado por meio do comando:

```
mrec_train -n4 --input_format tsv --train "splits/dataset.tsv.train.*" --outdir models"
```

Após o término da execução, os modelos treinados são salvos no diretório *models* para serem utilizados na geração e avaliação de recomendações, ambas concebidas por meio do comando a seguir:

```
mrec_predict --input_format tsv --test_input_format tsv --train "dataset.tsv.train.*" --modeldir models --outdir recs"
```

Os modelos obtidos no diretório *models* são usados para criar e avaliar as recomendações. Ao término da execução, os resultados da avaliação das recomendações são apresentados. Como mencionado anteriormente, informações detalhadas sobre o uso de parâmetros e outras opções disponíveis na preparação de dados de treino, treinamento de recomendadores ou geração e avaliação de recomendadores estão disponíveis na documentação do *mrec* (MREC, 2013).

2.6 PRINCIPAIS VANTAGENS E DESVANTAGENS

Nesta seção são apresentadas as principais vantagens e desvantagens do *mrec*.

2.6.1 VANTAGENS

Conforme mencionado, o funcionamento do *mrec* segue o paradigma híbrido de sistemas de recomendação. Com a disponibilidade dos cinco algoritmos citados na Seção 2.1, o *mrec* consegue gerar e avaliar recomendações com praticidade por meio de alguns comandos de execução no terminal, como descrito na Seção 2.5.

Por se tratar de um *software* de código-livre, qualquer usuário pode alterar funcionalidades existentes ou contribuir para a correção de *bugs*. Empresas podem integrar o *mrec* com os seus serviços sem muitas dificuldades uma vez que os dados de entrada e saída são arquivos que podem ser facilmente extraídos de uma base de dados e formatados para o padrão que o *mrec* reconhece. Como os comandos do *mrec* podem ser executados pelo terminal, é possível também a criação de *scripts* de recomendação para servidores passando os arquivos de dados formatados como parâmetro, além de permitir a inserção de novos algoritmos e novos comandos de execução.

Em adição, o *mrec* utiliza o *IPython* para treinar modelos e gerar recomendações em paralelo; e possui um exemplo na documentação sobre como executá-lo em um *cluster* de computadores utilizando o programa *StarCluster* em conjunto com o *IPython*.

2.6.2 DESVANTAGENS

A ausência de suporte e divulgação acarretou na sua descontinuação e possivelmente na sua remoção do gerenciador de pacotes *pip*. Mesmo que a última atualização do seu repositório tenha ocorrido em 2016, o programa em si estagnou na versão 0.3.0, lançada em 2013.

A documentação está desatualizada e, portanto, pode induzir a erros no momento de instalação. Apesar de possuir um tutorial de uso e listagem dos módulos, funções e comandos, faltam exemplos e detalhes mais específicos sobre os seus algoritmos e métricas de avaliação.

3. EXPERIMENTO REALIZADO

Nesta seção é avaliado o desempenho do mrec por meio da execução de um experimento que envolve as suas principais tarefas e segue as indicações de uso da documentação do mrec. O algoritmo escolhido para a geração e treinamento de modelos foi o SLIM, que é a opção padrão de execução. No experimento foi utilizado o conjunto de dados *MovieLens 100K*². Este, por sua vez, consiste em 100.000 avaliações de 1000 usuários para 1700 filmes, além de informações sobre gênero dos filmes e dados sobre os usuários que estão contidos em arquivos de texto organizados em formato tsv:

u.user: Todos os usuários e suas informações pessoais;

u.occupation: Todas as ocupações (profissões) dos usuários do conjunto de dados;

u.info: O total de usuários, filmes e avaliações que estão no conjunto de dados;

u.item: Todos os itens (filmes) que estão no conjunto e seus respectivos dados.

u.genre: Todos os gêneros de filmes que estão no *MovieLens*.

u.data: Tabela de avaliações usuário-item.

Neste experimento, como foi utilizado o algoritmo SLIM, a única entrada necessária foi a **u.data**, que consiste dos seguintes dados ordenados em colunas: código do usuário, código do filme, nota que o usuário forneceu ao filme e o *timestamp*. Os outros arquivos não foram necessários para a realização do experimento.

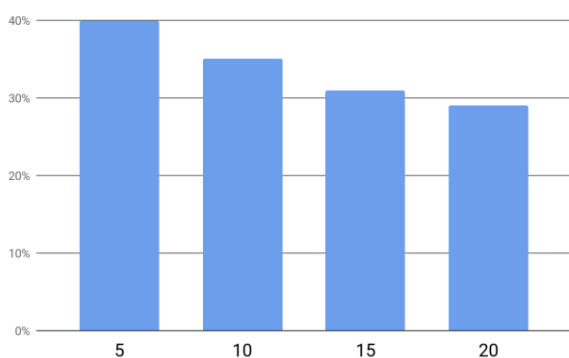
O experimento foi executado em uma máquina com o sistema operacional Os X v10.14.5. A instalação e o uso do mrec ocorreu com base nos passos definidos na Seção 2.5, onde são descritos os comandos para os processos de preparação do conjunto de dados, geração e treino de modelos, predição e também avaliação de recomendações.

Na etapa de preparação do conjunto de dados foram usados parâmetros para separar todos os filmes que os usuários deram notas iguais ou superiores ao valor 4 em subconjuntos de treino e teste de mesmo tamanho. As notas feitas pelos usuários foram substituídas pelo valor 0 caso fossem menor que 4, ou 1 se fossem maior.

Os parâmetros definidos no comando de execução da fase de geração e treino de modelos permaneceram praticamente idênticos aos apresentados na Seção 2.5. O mesmo pode ser dito em relação à fase de criação e avaliação de recomendações.

O Gráfico 1 mostra a informação de *Precision@k*, onde *k* representa os valores 5, 10, 15 e 20. Por exemplo, o valor de precisão *Precision@5* presente no Gráfico 1, indica que a cada 5 filmes recomendados, são encontrados em média dois deles no subconjunto de teste, ou seja, o recomendador sugeriu dois filmes que o usuário gostou e avaliou positivamente.

Gráfico 1. Precisão para 5, 10, 15 e 20 recomendações.



O gráfico também mostra que a precisão das recomendações diminui conforme o número de itens recomendados aumenta, o que pode ser um problema para recomendações de muitos itens onde a predição tem mais chances de não estar correta.

Após o término da execução do mrec foram obtidas as recomendações geradas diretamente pelo modelo treinado. Esses dados são exibidos no terminal após a execução como uma lista contendo o identificador de usuário, identificador do filme e o *score* de predição para as melhores recomendações. Os valores do *score* são organizados em ordem decrescente e, conforme apresentado na Tabela 1, o algoritmo entendeu que o usuário tem interesse pelos gêneros drama, ação ou aventura e recomendou filmes que

² <https://grouplens.org/datasets/movielens/latest>

contém um ou mais destes gêneros. Levando em consideração que o usuário para qual foram feitas as recomendações avaliou positivamente filmes como Amadeus (1984) e Jejum de Amor (1940) é possível perceber que o sistema acertou ao identificar o seu interesse por filmes de drama, e sugeriu filmes que se encaixam nesse e em outros gêneros com chances relativamente boas de aceitação, por exemplo o filme Fargo (1996) que se encaixa no gênero de drama e também drama policial foi recomendado com uma probabilidade de 38,2% de ser bem avaliado pelo usuário.

Tabela 1. Predição do *score* de preferência.

Filme	Score
Fargo (1996)	38,2%
Os Caçadores da Arca Perdida (1981)	27,5%
Golpe de Mestre (1973)	20,6%
A Lista de Schindler (1993)	20,2%
Um Estranho no Ninho (1975)	19,0%
O Exterminador do Futuro (1984)	18,8%
Intriga Internacional (1959)	16,8%
A Primeira Noite de um Homem (1967)	16,7%
O Retorno de Jedi (1983)	16,6%
Todos a Bordo (1996)	16,4%

Com o experimento, é possível verificar que as ações necessárias para a sua execução foram relativamente simples de realizar. Em adição, o tempo de processamento do experimento foi curto, e os resultados obtidos foram satisfatórios e devidamente armazenados após o término do processamento. Portanto, é possível afirmar que o sistema é capaz de realizar as suas principais tarefas (preparação, treino, predição e avaliação) com robustez.

4. CONSIDERAÇÕES FINAIS

Durante o desenvolvimento deste trabalho, dados sobre as características, uso e funcionamento do mrec foram reunidos e descritos, enquanto um experimento para avaliação de desempenho foi realizado. A análise dos resultados obtidos no experimento em união com as informações analisadas na Seção 2 permitiram a definição de um veredito para o mrec.

Ao analisar a Seção 3, é possível afirmar que o sistema executa com robustez as funcionalidades esperadas de um sistema de recomendação, já que é capaz de gerar recomendações de qualidade e possui boas medidas para a avaliação das mesmas. Em adição, ainda fornece o recurso de paralelismo para o treinamento de modelos e avaliação de recomendações.

Entretanto, o mrec já não possui suporte por parte dos desenvolvedores. Desde 2013 que o mesmo continua na versão 0.3.1 e, conseqüentemente, a sua documentação também encontra-se desatualizada. Ademais, por ter sido removido dos repositórios do gerenciador de pacotes pip, é altamente provável que o seu desenvolvimento foi descontinuado.

Mesmo sendo um sistema de recomendação descontinuado, foi possível perceber que ele possui funcionalidades que o destacam de sistemas de recomendação semelhantes como Pandas, Scikit-Surprise ou CaseRecommender, por exemplo a sua forma de integração, que não obriga o programador a alterar o software que está desenvolvendo para utilizar o mrec, ao invés disso é possível executar o mrec

separadamente e obter apenas os resultados das recomendações, isso faz com que os softwares a serem integrados com o mrec sejam menos dependentes dele e tenham uma manutenibilidade melhor.

Os fatores negativos não inviabilizam o seu uso para qualquer finalidade ou a sua integração com outros sistemas. O mrec ainda é um sistema de recomendação prático, robusto e de fácil usabilidade. O código-fonte e a documentação, embora datados e sem suporte, continuam funcionais e disponíveis para qualquer indivíduo cujas necessidades possam ser correspondidas pelo sistema.

Dado o custo de projetar e desenvolver um sistema de recomendação e de acordo com o conhecimento adquirido durante o experimento, ao invés de desenvolver um novo sistema com as características e vantagens do mrec, é mais vantajoso que ele seja atualizado para novas versões do python e usado em meios acadêmicos ou comerciais, se tornando uma boa opção para programadores que utilizam do recurso de recomendações em seus softwares.

REFERÊNCIAS

DUDA, Richard O.; HART, Peter E.; STORK, David G. **Pattern classification**. John Wiley & Sons, 2012.

HU, Y.; KOREN, Y.; VOLINSKY, C. **Collaborative filtering for implicit feedback datasets**. Citeseer, 2008. <https://doi.org/10.1109/ICDM.2008.22>

JANNACH, D.; ZANKER, M.; FELFERNIG, A.; FRIEDRICH, G. **Recommender systems: an introduction**. Cambridge University Press, 2010. <https://doi.org/10.1017/CBO9780511763113>

MOVIELENS. **MovieLens Dataset** <https://grouplens.org/datasets/movielens/>, 2019, accessed: 2019-08-22.

MREC. **mrec recommender systems library** <http://mendeley.github.io/mrec/>, 2013, accessed: 2019-06-11.

NING, X; KARYPIS, G. **Slim: Sparse linear methods for top-n recommender systems in 2011 IEEE 11th International Conference on Data Mining**. IEEE, 2011, pp. 497–506. <https://doi.org/10.1109/ICDM.2011.134>

PÉREZ, F.; GRANGER, B. E. **IPython: a system for interactive scientific computing**. Computing in Science and Engineering, vol. 9, no. 3, pp. 21–29, May 2007. [Online]. Available: <https://ipython.org>. <https://doi.org/10.1109/MCSE.2007.53>

WESTON, J.; BENGIO, S.; USUNIER, N. **Large scale image annotation: learning to rank with joint word-image embeddings**. Machine learning, vol. 81, no. 1, pp. 21–35, 2010. <https://doi.org/10.1007/s10994-010-5198-3>

XIAO, B.; BENBASAT, I. **E-commerce product recommendation agents: use, characteristics, and impact**. MIS quarterly, vol. 31, no. 1, pp. 137–209, 2007. <https://doi.org/10.2307/25148784>

RESUMOS DE PESQUISA

APLICAÇÃO DE MÉTODOS DE APRENDIZADO DE MÁQUINA NA DESCOBERTA DE PERFIL DE ALUNOS.....	11
AUXÍLIO NO TRATAMENTO DE PACIENTES COM HIPERTENSÃO E DIABETES FAZENDO USO DE REALIDADE VIRTUAL	12
CLASSIFICAÇÃO DE FACES HUMANAS USANDO REDE NEURAL CONVOLUCIONAL.....	13
DETECÇÃO E RECONHECIMENTO DE FACES UTILIZANDO VISÃO COMPUTACIONAL E ALGORITMOS DE APRENDIZADO DE MÁQUINA	14
LPSVERSION: IMPLEMENTAÇÃO DE CONTROLE DE VERSÃO.	15
O USO DE REINFORCEMENT LEARNING NOS JOGOS DIGITAIS.....	16
PARKINSON'S TREATMENT SYSTEM - APLICAÇÃO IMERSIVA COM REALIDADE VIRTUAL PARA O AUXÍLIO NO TRATAMENTO DE PACIENTES COM A DOENÇA DE PARKINSON	17
SIMULADOR EM REALIDADE VIRTUAL PARA AJUDAR NO TRATAMENTO DE TRANSTORNO DE FOBIA DE DIREÇÃO	18
UM FRAMEWORK PARA RECONHECIMENTO DE GESTOS EM LIBRAS A PARTIR DE IMAGENS DE VÍDEO.....	19
UMA BIBLIOTECA DE ROTINAS MATEMÁTICAS PROGRAMADAS EM GPU COM CUDA EMPACOTADAS EM UMA DLL.....	20
UTILIZANDO DEEP LEARNING PARA AUXILIAR O PROCESSO DE DETECÇÃO DE LEUCEMIA LINFOIDE AGUDA	21
VRSMOKING - UMA FERRAMENTA DE REALIDADE VIRTUAL E AUMENTADA APLICADA AO TRATAMENTO DE DEPENDENTES QUÍMICOS DO TABACO E DERIVADOS.	22

Pesquisa (ENAPI)

UNIVERSIDADE DO OESTE PAULISTA - UNOESTE

Comunicação oral

Ciências Exatas e da Terra

Ciência da Computação

APLICAÇÃO DE MÉTODOS DE APRENDIZADO DE MÁQUINA NA DESCOBERTA DE PERFIL DE ALUNOS

AUGUSTO CESAR SILVA CRUZ

HELTON MOLINA SAPIA

DANILLO ROBERTO PEREIRA

Instituições de ensino adquiriram a cultura de armazenar os dados de seus alunos, como: idade, histórico escolar, onde cursou o ensino médio e se possui formação anterior. A transformação destes dados em informação e possivelmente em conhecimento, faz com que o investimento na coleta e armazenamento agregue valor às instituições, descobrindo novas relações entre os dados que não se tinha conhecimento. Este trabalho tem como objetivo, realizar o estudo das informações obtidas por métodos de classificação e agrupamento não supervisionados, traçar uma relação entre a evasão, troca de curso e reprova dos alunos junto aos dados cedidos. Pretendeu-se antecipar quando um determinado aluno está mais suscetível à evasão. Foram solicitados dados a UNOESTE referentes aos alunos, como: sexo, idade, estado civil, cidade onde reside e histórico escolar. Após a coleta, os dados passaram por processos de limpeza e normalização, principalmente para que pudessem ser utilizados nos métodos de aprendizado. Os métodos utilizados foram: K Nearest Neighbor (KNN), Support Vector Machine (SVM) e Naive Bayes. Cada um deles foi submetido a vários testes onde buscou-se atingir os melhores parâmetros para a comparação dos resultados. Após a aplicação dos métodos de aprendizado foi alcançada uma taxa de acerto de 93.87% com o método SVM utilizando-se apenas 40% da quantidade total de dados para treino. Enquanto os métodos KNN e Naive Bayes alcançaram apenas 70.54% e 15.05% respectivamente, com a mesma parcela de dados para teste. Através de uma análise manual dos dados foi observado que, entre os alunos que desistiram, aproximadamente, 55% deles possuem a bolsa PROUNI. Mas entre os alunos bolsistas, apenas 18% chega a desistir. Após a aplicação dos métodos de aprendizado de máquina, o método SVM mostrou-se melhor para indicar quando um aluno está mais propenso a desistir do curso. Com a análise manual dos dados foi comprovado que não existe um fator único que caracteriza o aluno que pode ou não desistir do curso. Isso reforçou a necessidade de métodos de aprendizado de máquina, uma vez que eles analisaram os dados sem qualquer viés prévio.

Pesquisa (ENAPI)

UNIVERSIDADE DO OESTE PAULISTA - UNOESTE

Ciências Exatas e da Terra

Comunicação oral

Ciência da Computação

AUXÍLIO NO TRATAMENTO DE PACIENTES COM HIPERTENSÃO E DIABETES FAZENDO USO DE REALIDADE VIRTUAL

NATHALIA RODRIGUES FONSECA
ROBSON AUGUSTO SISCOOTTO
ANA PAULA COELHO FIGUEIRA FREIRE
FRANCIS LOPES PACAGNELLI

O manejo de doenças como Hipertensão Arterial Sistêmica e a Diabetes Mellitus são uns dos principais desafios para saúde pública no Brasil. Ambas doenças relacionadas com altos índices de morbidade e mortalidade mundial. Sabe-se que a maior parte de internações de pacientes com essas patologias poderiam ter sido evitadas, levando em consideração que o motivo do agravamento da doença é a falta de cuidados que cada patologia necessita, a informação e conscientização dos cuidados com a doença é crucial para o tratamento. Este trabalho apresenta uma aplicação de Realidade Virtual interativa focada na reabilitação de pacientes com Hipertensão Arterial Sistêmica e Diabetes Mellitus. Este trabalho é uma parceria com o grupo HIPERDIA (Sistema de Cadastro e Acompanhamento de Hipertensos e Diabéticos) que atende, em posto de saúde, pacientes com essas patologias e aplica atividades individuais e/ou dinâmicas em grupo a cada visita do paciente. A aplicação permite dar continuidade às atividades de forma online e interativa. A metodologia utilizada é identificada como uma abordagem qualitativa e quantitativa, iniciada com a revisão sistemática e trabalhos relacionados, pelos quais foram definidos os requisitos funcionais para o desenvolvimento da aplicação. Para o desenvolvimento do projeto foram utilizados celulares, computadores e óculos RV. Diante disso, um protótipo foi desenvolvido, testado e avaliado qualitativamente. Os testes foram realizados com profissionais da área da fisioterapia e computação por meio de questionários sobre a usabilidade da aplicação, fazendo uso de 15 perguntas, sendo atribuída uma nota de 1 a 5. Vale lembrar que testes com pacientes não foram realizados, tais testes serão realizadas em projetos futuros. Como resultados parciais, a aplicação foi qualificada adequadamente e aprovada pelos profissionais. Com a sua conclusão, esta será disponibilizada para os monitores do programa HIPERDIA, os quais serão responsáveis pela interação do aplicativo com o paciente em trabalhos futuros. or meio desses resultados positivos podemos concluir a viabilidade da aplicação. Além disso, com a conclusão da aplicação, serão escritos artigos e submetidos a eventos da área. Órgão de fomento financiador da pesquisa: Coordenadoria de Pesquisa, Desenvolvimento e Inovação (CPDI). 4984

Pesquisa (ENAPI)

UNIVERSIDADE DO OESTE PAULISTA - UNOESTE

Ciências Exatas e da Terra

Comunicação oral

Ciência da Computação

CLASSIFICAÇÃO DE FACES HUMANAS USANDO REDE NEURAL CONVOLUCIONAL

PAULO VITOR REFATTI
ALMIR OLIVETTE ARTERO
FRANCISCO ASSIS DA SILVA
DANILLO ROBERTO PEREIRA

A classificação de imagens é uma tarefa de grande importância, sendo muito usada no monitoramento de ambientes para identificar pessoas e objetos, identificar automóveis em rodovias, dentre outras aplicações. O processo mais usado para a classificação consiste em extrair diversos descritores das imagens e, em seguida, usar alguma técnica de classificação, como por exemplo, uma rede neural ou mesmo o classificador Bayesiano. Porém, a maior dificuldade com esta metodologia consiste em se definir quais são os melhores descritores a serem extraídos das imagens. Uma abordagem mais recente sugere o uso de uma rede neural convolucional (CNN - convolutional neural networks), que dispensa a necessidade de se extrair os descritores das imagens. Neste caso, a própria rede extrai as informações diretamente das imagens. O objetivo deste trabalho consiste em avaliar o desempenho de uma rede neural convolucional, na classificação de imagens de faces humanas. As redes neurais estão sendo implementadas usando a linguagem Python, que disponibiliza diversas funções que simplificam a implementação de CNNs. Nos experimentos, estão sendo avaliadas diferentes configurações de redes, sendo usadas, entre outras, as imagens de faces disponíveis na base The Japanese Female Facial Expression (JAFFE), disponível para livre acesso através do endereço eletrônico <http://www.kasrl.org/jaffe.html>. O uso de imagens largamente usadas pela comunidade científica é recomendável, porque permite realizar uma comparação quantitativa dos resultados obtidos neste trabalho em relação à aplicação de outras técnicas mais tradicionais, usando a extração de características das imagens. Os resultados obtidos até o momento mostram que a CNN consegue realizar uma classificação bem satisfatória. Para a avaliação dos resultados, serão feitas comparações com os resultados obtidos com outras técnicas, como por exemplo, o LPB (Local binary patterns). O uso de redes neurais convolucionais em diversas áreas tem apresentado bons resultados em relação à metodologia tradicional, assim, neste trabalho também são esperados bons resultados. Espera-se também que a experimentação de várias estruturas da CNN possam revelar quais parâmetros são mais importantes para a aplicação específica de identificação de faces humanas.

Pesquisa (ENAPI)

UNIVERSIDADE DO OESTE PAULISTA - UNOESTE

Ciências Exatas e da Terra

Comunicação oral

Ciência da Computação

DETECÇÃO E RECONHECIMENTO DE FACES UTILIZANDO VISÃO COMPUTACIONAL E ALGORITMOS DE APRENDIZADO DE MÁQUINA

LUCAS JOSÉ DA COSTA
THIAGO LUZ DE SOUSA
FRANCISCO ASSIS DA SILVA
MÁRIO AUGUSTO PAZOTI
LEANDRO LUIZ DE ALMEIDA
ALMIR OLIVETTE ARTERO
DANILLO ROBERTO PEREIRA

Tecnologias de detecção e reconhecimento facial estão sendo muito utilizadas em várias áreas distintas como para melhoria de sistemas de segurança, aplicações de entretenimento, e para fins comerciais. Existem várias técnicas de reconhecimento de faces implementadas, no entanto ainda há muito espaço para novas ideias e melhorias das que já existem, além de problemas a serem resolvidos, como variação de iluminação do ambiente, mudança das características físicas do rosto, variação de equipamentos utilizados e resolução das imagens utilizadas. Objetiva-se realizar uma análise comparativa entre os métodos de detecção e reconhecimento facial assim como o tempo de execução dos mesmos. Foram utilizados os algoritmos de reconhecimento facial Eigenface, Fisherface e LBPH em conjunto com o algoritmo de detecção facial Haar Cascade, todos da biblioteca OpenCV. Assim como o algoritmo CNN para reconhecimento facial em conjunto com o algoritmo de detecção facial HOG, estes da biblioteca Dlib. Os datasets de imagens de faces públicos utilizados foram o Yale com 165 imagens, Georgia Tech com 750 imagens e o Caltech com 445 imagens. Os algoritmos foram executados em um computador com 8GB de memória RAM, processador Intel Core I5 3ª geração e SSD 240 GB. Os resultados obtidos com o algoritmo CNN foram de 100% de acerto com os datasets Caltech e Georgia Tech com tempo de execução de 10.5 e 16.1 segundos respectivamente, e 93% de acerto e tempo de 2.4 segundos como dataset Yale. O LBPH obteve 88% de acerto com o dataset Caltech, 85% com Georgia Tech, e 73% com Yale, com tempos de execução de 1.5, 1.8 e 1.6 segundos respectivamente. O Fisherface obteve 96% de acerto com o dataset Yale, 80% com Caltech, e 56% com Georgia Tech, com tempos de execução de 1.4, 1.8 e 2.3 segundos respectivamente. O Eigenface obteve 83% de acerto com o dataset Yale, 70% com Georgia Tech, e 67% com Caltech, com tempos de execução de 2.7, 19.5 e 18.8 segundos respectivamente. Embora o algoritmo CNN tenha apresentado mais acertos que os demais, ainda existem outros fatores que devem ser levados em consideração como grau de confiabilidade e tempo de execução para escolha de qual algoritmo será utilizado. Em Visão Computacional, já foram feitas várias pesquisas para desenvolvimento de algoritmos para detecção e reconhecimento de faces. Este trabalho contribui com a análise de eficiência de alguns destes métodos, acrescentando que a escolha entre eles será ao que melhor atende os requisitos necessários.

Pesquisa (ENAPI)
Comunicação oral

UNIVERSIDADE DO OESTE PAULISTA - UNOESTE
Ciências Exatas e da Terra
Ciência da Computação

LPSVERSION: IMPLEMENTAÇÃO DE CONTROLE DE VERSÃO.

VINICIUS SOUZA VASCONCELOS DOS SANTOS
FRANCISCO VIRGINIO MARACCI
MÁRIO AUGUSTO PAZOTI
FRANCISCO ASSIS DA SILVA

Ao construir um software, este passa por uma série de mudanças desde a sua concepção até a implantação. Durante este processo de desenvolvimento isto deve ser gerenciado e organizado para haver um número menor de falhas e problemas futuros. A gerência de configuração de software (GCS) cuida desta área. Existem várias ferramentas que cuidam disto, um problema encontrado é que muitas vezes a heterogeneia e a falta de comunicação ou compartilhamento entre algumas ferramentas. Após uma pesquisa a fundo do que estas ferramentas proporcionam percebe-se que ainda há pontos a serem tratados. Um destes pontos é que nenhuma destas ferramentas consegue saber exatamente o que um componente tem de diferente de outro em um repositório. Isso torna o controle de versão, mudança e a entrega um pouco menos preciso pensando em uma LPS. Esta pesquisa propõe a criação de uma ferramenta que além de fazer o versionamento de um projeto, auxilie em dizer qual das componentes é a mais adequada para incorporar no novo projeto ou qual das versões contidas no repositório é a mais próxima do que se necessita de acordo com suas variabilidades. Então a pesquisa tem por objetivo o desenvolvimento de uma ferramenta de GSC que por intermédio do uso de modelo de características (Feature Model), modele os aspectos comuns e variabilidades em uma LPS tornando o reuso de artefatos ágil. A primeira fase consiste do estudo dos assuntos. Logo após, parte-se para a implementação do protótipo da ferramenta. A próxima fase consiste em implementar a ferramenta utilizando o modelo entidade relacionamento e diagrama de classes. Ao final haverá testes na ferramenta e o artigo constará todos os estudos realizados e resultados obtidos. Os resultados obtidos até o presente momento são, uma melhor organização dos repositórios quando aplicados em uma LPS, o reuso inteligente de artefatos de uma mesma família de softwares e agilidade, praticidade em buscar e versionar componentes no repositório para construir um sistema baseado em componentes. Bons resultados quando se pensa na construção de grandes sistemas. Sistemas com grandes volume de componentes e necessidade de reuso inteligente e rápido, assim agilizando o trabalho e não permitindo o retrabalho de forma desnecessária. Algumas metas já foram cumpridas até o presente momento, assim atingindo o objetivo final do projeto que é propor e implementar uma ferramenta de auxílio na GSC. Seguimos na segunda parte que são os teste e a finalização do artigo.

Pesquisa (ENAPI)

UNIVERSIDADE DO OESTE PAULISTA - UNOESTE

Ciências Exatas e da Terra

Comunicação oral

Ciência da Computação

O USO DE REINFORCEMENT LEARNING NOS JOGOS DIGITAIS

HENRIQUE RAMOS RICCI
BRUNO VIANA DA SILVA
MÁRIO AUGUSTO PAZOTI
FRANCISCO ASSIS DA SILVA
LEANDRO LUIZ DE ALMEIDA
ALMIR OLIVETTE ARTERO
DANILLO ROBERTO PEREIRA

Como o aumento do poder computacional, estudos na área de inteligência artificial têm crescido fortemente em todo o mundo e surpreendido a todos com seus resultados, seja em carros autônomos, em algoritmos que conseguem vencer humanos em jogos de tabuleiro, até aplicações do dia a dia como assistentes virtuais para bancos, sugestão de músicas ou vídeos em aplicativos ou jogos em geral. Dentre as áreas da IA, o aprendizado por reforço se apresenta como uma estratégia interessante, seguindo a filosofia de que um agente pode aprender com suas tentativas e erros, assim como um ser vivo aprende que o fogo a uma certa distância aquece, mas quando se chega muito perto, queima. Pensando nisso, há uma necessidade em avaliar o desempenho desses algoritmos, averiguando se suas aplicações em tomadas de decisões reais podem ser precisas a ponto de não gerar falhas que comprometam a sua aplicação. O objetivo do projeto é analisar os resultados do algoritmo deep q-learning aplicado ao jogo Ms.PacMan. Assim como na vida real, um jogo possui inúmeros estados diferentes, em que ações devem ser tomadas de maneira a maximizar os resultados ao fim da aplicação. Ao conseguir vencer uma fase do jogo sem falhas, compreende-se que aplicar o algoritmo no mundo real, em um carro autônomo por exemplo, trará resultados similares, uma vez que o jogo pode ser uma simulação do mundo. Ao final do projeto, espera-se que o algoritmo consiga vencer uma fase do jogo de forma hábil. Caso esse resultado não seja atingido, um levantamento será feito para encontrar o que pode estar gerando a falha e aplicar a correção. Para medir a eficiência e eficácia do algoritmo, além de testes com variação nos parâmetros mais importantes do algoritmo, será utilizado uma nova metodologia para avaliação de desempenho de algoritmos baseados em aprendizagem por reforço. Para melhorar o processo de treinamento, está sendo utilizado um processamento em GPU, conhecido como CUDA. Até o presente momento, o algoritmo conseguiu concluir cerca de 45% de uma fase, porém os resultados tendem a aumentar à medida que mais horas de treinamento sejam realizadas. Com esses resultados, nota-se que o algoritmo está evoluindo e tende a alcançar os resultados esperados. Como o processamento de algoritmos de inteligência artificial é algo extremamente custoso, os treinamentos levam muito tempo para mostrarem os resultados. Nessa nova fase de teste, com o aumento do poder computacional, espera-se resultados melhores em menos tempo.

Pesquisa (ENAPI)

UNIVERSIDADE DO OESTE PAULISTA - UNOESTE

Ciências Exatas e da Terra

Comunicação oral

Ciência da Computação

PARKINSON'S TREATMENT SYSTEM - APLICAÇÃO IMERSIVA COM REALIDADE VIRTUAL PARA O
AUXÍLIO NO TRATAMENTO DE PACIENTES COM A DOENÇA DE PARKINSON

YAGO RAFAEL COUTINHO DE MENEZES
ROBSON AUGUSTO SISCOOTTO
RENATA APARECIDA DE OLIVEIRA LIMA
FRANCIS LOPES PACAGNELLI
MARIANE PEREIRA DOS REIS SANTOS
RUBIA STEFANY MOREIRA GALVAO
ERSHILEY DA SILVA LIMA
TAMIRES DA SILVA FLORIANO BARBOSA

A Doença de Parkinson (DP) é um distúrbio progressivo e crônico do sistema nervoso central que afeta principalmente o cérebro. Apesar de ser incurável, vários tratamentos são possíveis de serem aplicados com o intuito de amenizar os danos motores causados e retardar o seu avanço no indivíduo. Recentemente, o uso de intervenções tecnológicas aplicadas a área da saúde vem ganhando notoriedade através de sistemas imersivos em Realidade Virtual (RV) ou Aumentada (RA) e sistemas adaptativos. Entretanto, devido à recente novidade destas tecnologias, poucos programas de reabilitação baseados na imersão adaptados especificamente à DP existem no mercado. Por não haver essas aplicações, os profissionais fisioterapeutas costumam adaptar o tratamento a jogos interativos limitando assim os resultados, uma vez que os movimentos destes jogos não foram projetados com a finalidade de auxiliar no tratamento de pessoas com dificuldades motoras. Este trabalho apresenta o desenvolvimento de uma aplicação de RV voltada para o auxílio no tratamento da DP, através do uso de ambientes virtuais tridimensionais voltados à exercícios controlados de fisioterapia. A metodologia iniciou-se por meio de uma revisão sistemática objetivando identificar trabalhos correlatos e suas tecnologias. Em seguida, reuniões com os fisioterapeutas foram realizadas para coleta dos requisitos. Posteriormente, um protótipo inicial foi desenvolvido utilizando os recursos disponíveis nos laboratórios da Faculdade de Informática de Presidente Prudente, tais como computadores, entre outros. Por não haver a submissão de pacientes reais ao protótipo, através de uma adaptação da heurística de usabilidade de Nielsen testes qualitativos foram aplicados por meio de questionários com 15 perguntas. Os pesquisadores e profissionais pontuaram cada questão com valor de 1 a 5. Ao final da análise dos resultados, a aplicação foi qualificada com um resultado parcial de 4.4, sendo considerada adequada e aprovada. Com a conclusão da aplicação, esta poderá ser aplicada em sessões fisioterapêuticas na clínica local da Universidade do Oeste Paulista durante a rotina de tratamento a DP. Por meio desses resultados positivos podemos concluir a viabilidade da aplicação adaptada ao tratamento da DP disponibilizando atividades físicas fundamentadas em um programa real de tratamento; além de fundamentar a escrita e a submissão de artigos aos eventos da área. Órgão de fomento financiador da pesquisa: Coordenadoria de Pesquisa, Desenvolvimento e Inovação (CPDI, protocolo núm. 4993).

Pesquisa (ENAPI)

UNIVERSIDADE DO OESTE PAULISTA - UNOESTE

Ciências Exatas e da Terra

Comunicação oral

Ciência da Computação

SIMULADOR EM REALIDADE VIRTUAL PARA AJUDAR NO TRATAMENTO DE TRANSTORNO DE FOBIA DE DIREÇÃO

IGOR ANDRADE BRITO
ROBSON AUGUSTO SISCOOTTO
KARINE AMARAL MAGALHÃES
ANA PAULA DOMENEGHETTI PARIZOTO FABRIN

A fobia de dirigir é caracterizado pelo medo irracional de alguma coisa ou situação e é um problema muito recorrente no mundo atual e atrapalha a vida de muitas pessoas. Uma forma comum de tratar esse tipo de problema é a exposição in vivo, ou seja, o paciente é colocado para dirigir um carro para perder seu medo aos poucos, porém, ao ser colocado no trânsito real, o paciente é exposto a alguns riscos que não podem ser controlados durante o tratamento, como por exemplo acidentes, podendo até causar fatalidades. O uso de simuladores vem ajudando muitas pessoas com esse tipo de transtorno, principalmente, sem gerar riscos reais a integridade da pessoa e proporcionando uma maior confiabilidade na hora de pegar um carro. Dentro desse contexto, a realidade virtual pode contribuir para simuladores mais realista e motivacionais, contribuindo para uma experiência da exposição in vivo mas sem ter que passar pelos perigos do trânsito. Este trabalho apresenta o desenvolvimento de software computacional para auxiliar no tratamento de transtornos de fobia de dirigir por meio de ambientes virtuais e realidade virtual. A metodologia desenvolvida iniciou-se por uma revisão sistemática, trabalhos correlatos e tecnologias. Além disso, reuniões com os profissionais da psicologia foram realizadas para a coleta dos requisitos e definição do escopo. Posteriormente, um protótipo inicial foi desenvolvido utilizando os recursos disponíveis nos laboratórios da Faculdade de Informática de Presidente Prudente, tais como computadores, volante e câmbio de trocar marchas utilizados em jogo visando simular um carro, um óculos de realidade virtual em conjunto com um celular para a experiência imersiva. Os testes foram realizados com profissionais da área da psicologia e computação por meio de questionários sobre a usabilidade da aplicação, baseado na Heurística de Nielsen. Foram utilizadas 15 perguntas, sendo atribuída uma nota de 1 a 5. Vale lembrar que testes com pacientes não foram realizados, tais testes serão realizadas em projetos futuros. Como resultados parciais, a aplicação foi qualificada adequadamente e aprovada pelos profissionais. Com a sua conclusão, esta poderá ser aplicada em sessões na clínica local da Universidade do Oeste Paulista. Por meio desses resultados positivos, pode-se concluir a viabilidade da aplicação. Além disso, um artigo científico a respeito da pesquisa e com os resultados obtidos será escrito.

Pesquisa (ENAPI)

UNIVERSIDADE DO OESTE PAULISTA - UNOESTE

Ciências Exatas e da Terra

Comunicação oral

Ciência da Computação

UM FRAMEWORK PARA RECONHECIMENTO DE GESTOS EM LIBRAS A PARTIR DE IMAGENS DE VÍDEO

LUCAS BARBOZA SANCHES

SILVIO ANTONIO CARRO

Atualmente o surgimento de tecnologias direcionadas à inserção digital de deficientes auditivos tornam as desigualdades sociais para este público cada vez menos visíveis. Tem-se como exemplos os projetos: VLibras, Hand Talk, uSound e ProDeaf. A contribuição do presente estudo é fornecer tecnologia de fácil acoplamento, na forma de framework, para o desenvolvimento de aplicativos com recursos direcionados à inserção digital de deficientes auditivos. Sendo assim, contribuirá com o surgimento de novos produtos na área. O framework escrito em linguagem Java, é capaz de realizar o reconhecimento de sinais da Língua Brasileira de Sinais, capturados em vídeos e traduzidos para frases ou palavras equivalentes na língua portuguesa. O reconhecimento é realizado via técnicas e métodos de classificação da área de Learnig Machine. No desenvolvimento do projeto foi realizado uma sequência de estudos relacionados à estrutura gramatical da Língua Brasileira de Sinais, como a estruturação de frase e parâmetros, em seguida foi realizada a obtenção de conhecimentos referentes a área de Inteligência Artificial, aprofundando na subárea de Aprendizado de Máquina até atingir conceitos referentes às Redes Neurais Convolucionais (ConvNet ou CNN), YOLO (You only look once) e o uso biblioteca Deeplearning4j para construção de uma Rede Neural Convolucional. Para o protótipo inicial foi necessário a utilização de um DataSet de arquivos de imagens. O DataSet utilizado passou por modificações resultando em 2640 imagens com dimensões 50X50, estando estes separados em "Treino" e "Teste", possuindo cada 22 classes (A,B,C,D,E,F,G,I,L,M,N,O,P,Q,R,S,T,U,V,W,Y,Z). Logo foi possível definir 2112 imagens para o conjunto de treino e 528 imagens para o conjunto de teste. A rede neural convolucional foi estruturada em 3 camadas convolucionais com saídas 256, 256 e 256 e 3 camadas de polling, sob 100 épocas, foi possível obter como resultado a acurácia de 89,77% e precisão de 89,51%. A pesquisa demanda o estudo e a utilização de recursos tecnológicos de terceiros, testados e relatados em várias situações e por vários cientistas e no presente estudo utilizados na tradução da linguagem de sinais, extraídos em tempo real de vídeo capturados por dispositivos populares, como webcam ou câmeras de smartphones. Os resultados dependerão principalmente da contribuição de usuários desta tecnologia que compartilharão suas bases de testes com o projeto, criando assim datasets e treinos heterogêneos.

Pesquisa (ENAPI)

UNIVERSIDADE DO OESTE PAULISTA - UNOESTE

Ciências Exatas e da Terra

Comunicação oral

Ciência da Computação

UMA BIBLIOTECA DE ROTINAS MATEMÁTICAS PROGRAMADAS EM GPU COM CUDA
EMPACOTADAS EM UMA DLL

GABRIEL FRANCISCO RAMOS CASARO

ALMIR OLIVETTE ARTERO

FRANCISCO ASSIS DA SILVA

DANILLO ROBERTO PEREIRA

A GPU (Graphic Programming Unit) constitui uma tecnologia que já existe há algum tempo e, foi proposta para auxiliar a geração de gráficos e imagens, porém, devido ao seu grande poder de processamento, até maior que o do próprio processador (CPU), tem sido cada vez mais usada para o processamento de dados de uso geral. Com a popularização das GPUs, os programadores passaram a ter em suas máquinas um grande poder de processamento, porém, sem o uso de programas que acessam os recursos da GPU, todo o processamento continua sendo realizado apenas pela CPU, tornando o processamento mais demorado. Entretanto, a programação deste dispositivo ainda é uma tarefa complexa, envolvendo a instalação de bibliotecas específicas e a configuração correta dos compiladores. Além disso, as rotinas precisam ser implementadas paralelizando os processos. Assim, programadores menos experientes e usuários de outras áreas não conseguem usar o dispositivo. O objetivo deste trabalho consiste em simplificar o acesso dos recursos das GPUs aos programadores, que não possuem experiência com a programação paralela e a arquitetura CUDA (Compute Unified Device Architecture), por meio da disponibilização de uma biblioteca dinâmica DLL (Dynamic-link library), que ofereça um conjunto razoável de rotinas matemáticas, contemplando, inicialmente, operações envolvendo matrizes e vetores. As implementações das rotinas selecionadas estão sendo feitas em C++, usando o compilador NVCC e, empacotadas em uma DLL, que pode ser facilmente acessada por outros programas. Os resultados obtidos até o momento mostram um ganho razoável no tempo de processamento, em torno de oito vezes, em uma GPU com apenas 16 núcleos (Geoforce 8400 GS - 459 Mhz). Como o número de núcleos tem aumentado bastante nos últimos anos, em uma placa mais atual como a GTX 1050, com 640 núcleos, processando à 1354 MHz (custo aproximado de US\$150), deverão ser obtidos resultados bem superiores. Espera-se que a disponibilização desta biblioteca em forma de DLL possa contribuir para um uso mais massivo dos recursos da GPU, por parte de usuários com menor experiência em programação. Em trabalhos futuros deverão ser implementadas rotinas em outras áreas do conhecimento, como o processamento digital de imagens e outras áreas que demandam bastante processamento.

Pesquisa (ENAPI)

UNIVERSIDADE DO OESTE PAULISTA - UNOESTE

Ciências Exatas e da Terra

Comunicação oral

Ciência da Computação

UTILIZANDO DEEP LEARNING PARA AUXILIAR O PROCESSO DE DETECÇÃO DE LEUCEMIA LINFOIDE AGUDA

LEONARDO SANTOS MIRANDA

MÁRIO AUGUSTO PAZOTI

A leucemia é um tipo de câncer de sangue, podendo causar problemas no paciente. Esta doença pode ser do tipo aguda, quando ocorre a rápida proliferação das células, ocasionando em uma alta taxa de mortalidade. Visando diminuir os problemas que uma leucemia aguda pode causar, este projeto possui o objetivo de através do uso de redes neurais, na área de inteligência artificial, utilizando algoritmos provenientes de Deep Learning, obter uma maneira de realizar a detecção de forma mais precoce e precisa na leucemia, mais especificamente do tipo linfóide aguda. Como resultado esperado será proposto uma arquitetura da rede neural com uma alta acurácia para o reconhecimento de células precursoras da leucemia linfóide aguda. O presente projeto se justifica no sentido em que, amenizando os problemas citados anteriormente, a doença poderá ter um diagnóstico mais precoce, tratamento mais acelerado e, conseqüentemente, uma taxa de mortalidade ligeiramente menor. Este projeto tem como objetivo geral desenvolver uma rede neural treinada voltado para a análise de dados de diagnóstico da leucemia linfóide aguda visando auxiliar no diagnóstico precoce e precisão. Por meio da aplicação desenvolvida será possível analisar imagens de células sanguíneas, de tal forma que a rede seja capaz de reconhecer os padrões da imagem e classificar se é uma célula da leucemia linfóide aguda ou uma célula saudável, de acordo com o resultado final obtido da rede. A metodologia utilizada é identificada como uma abordagem qualitativa, que possui o objetivo de desenvolver uma rede neural treinada com o foco na área da saúde, mais especificamente na doença leucemia linfóide aguda, utilizando métodos provenientes da Inteligência Artificial, para que seja possível auxiliar o diagnóstico de forma mais precoce. Linguagem de programação Python 3.7 com o uso das bibliotecas Keras e TensorFlow. A base de dados foi divididos em dois grupos, sendo o primeiro separado com imagens celulares saudáveis e o outro com células da doença. Foram obtidos valores em torno de 97% de acurácia para ambos os casos. Trabalhos relacionados foram estudados e discutidos com o orientador. Para esta primeira etapa do projeto (Disciplina de Projeto I) foi feito um protótipo que classifica se dada a imagem celular, esta é pertencente da Leucemia Linfóide Aguda ou saudável. Para a etapa final, será proposto também a classificação do subtipos da doença (L1, L2 e L3).

Pesquisa (ENAPI)

UNIVERSIDADE DO OESTE PAULISTA - UNOESTE

Ciências Exatas e da Terra

Poster

Ciência da Computação

VRSMOKING - UMA FERRAMENTA DE REALIDADE VIRTUAL E AUMENTADA APLICADA AO
TRATAMENTO DE DEPENDENTES QUÍMICOS DO TABACO E DERIVADOS.

LUCAS PEREIRA DE LIRA
ROBSON AUGUSTO SISCOOTTO
ANA PAULA COELHO FIGUEIRA FREIRE
FRANCIS LOPES PACAGNELLI
BRUNA DE SOUZA GOULART
ISABELA MARTINS PERUQUE

Não é incomum nos dias atuais encontrarmos casos de pessoas com algum tipo de dependência química. Uma das mais comuns é a dependência à nicotina que atinge diversas pessoas em todo o mundo. Essa dependência é caracterizada pelo consumo regular do tabaco, de tal forma que o usuário não consiga ficar sem, acarretando-lhe dependência química sobre a nicotina presente no cigarro e derivados. A dependência química a nicotina é uma das mais severas devido a sua complexidade e seu consumo predispõe a diversos tipos de câncer, além de doenças cardíacas, hipertensão e outras condições. Portanto, a busca por um tratamento mais eficiente com novas estratégias complementares que possam auxiliar o dependente é uma necessidade, principalmente, comparados com as dificuldades encontradas nos tratamentos tradicionais ou, até mesmo, para iniciar um tratamento. Este projeto apresenta uma aplicação que faz uso de Realidade Virtual e Aumentada voltado ao tratamento de dependentes químicos da nicotina, fazendo uso de ambientes virtuais interativos, vídeos, áudios, textos, dentre outros, voltados a criação de sessões fisioterapêuticas elaboradas de acordo as necessidades do paciente. A metodologia utilizada é identificada como uma abordagem qualitativa e quantitativa que se iniciou com a elaboração de uma revisão sistemática a fim de identificar trabalhos relacionados. Para definição e coleta de requisitos foram realizadas reuniões com as partes interessadas. E com base nisso, um protótipo foi desenvolvido, testado e avaliado qualitativamente. Os testes foram realizados com profissionais da área da fisioterapia e computação por meio de questionários sobre a usabilidade da aplicação, baseado na Heurística de Nielsen. O questionário é constituído por 15 perguntas, sendo atribuída uma nota de 1 a 5. Vale lembrar que testes com pacientes não foram realizados, tais testes serão realizadas em projetos futuros. Os resultados parciais obtidos qualificaram a aplicação adequadamente e aprovada pelos profissionais. Com a sua conclusão, esta poderá ser aplicada em sessões fisioterapêuticas reais na clínica local da Universidade do Oeste Paulista. Por meio desses resultados positivos podemos concluir a viabilidade da aplicação para o tratamento de dependentes do tabagismo. Além disso, um artigo científico a respeito da pesquisa e com os resultados obtidos será escrito. Órgão de fomento financiador da pesquisa: Coordenadoria de Pesquisa, Desenvolvimento e Inovação (CPDI). Projeto: 4994.

RELATOS DE EXPERIÊNCIA

14ª PARTICIPAÇÃO DA FIPP NA MARATONA DE PROGRAMAÇÃO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO	24
15ª FESTA LINUX DA FACULDADE DE INFORMÁTICA DE PRESIDENTE PRUDENTE.....	25
15ª MARATONA DE PROGRAMAÇÃO DA FACULDADE DE INFORMÁTICA DE PRESIDENTE PRUDENTE - UNOESTE.....	26
2º HACKATHON FIPP - MARATONA DE ATIVIDADES TÉCNICAS DE COMPUTAÇÃO E INFORMÁTICA	27
32ª SEMANA DE COMPUTAÇÃO E INFORMÁTICA DA FIPP/UNOESTE - INFOESTE 2019.....	28
7º FIPPETEC - CICLO DE PALESTRAS E CURSOS DAS ETECS.....	29
PROGRAMANDO COM O SCRATCH: ENSINANDO LÓGICA DE PROGRAMAÇÃO E ALGORITMOS PARA ALUNOS INGRESSANTES DO ENSINO SUPERIOR.....	30
PROGRAMANDO O FUTURO.....	31
RECURSOS COMPUTACIONAIS PARA APOIO NO ENSINO DA DISCIPLINA DE ALGORITMO E LÓGICA DE PROGRAMAÇÃO	32

Ciência da Computação
Extensão (ENAEXT)
Comunicação oral

UNIVERSIDADE DO OESTE PAULISTA - UNOESTE
Ciências Exatas e da Terra
Ciência da Computação

14ª PARTICIPAÇÃO DA FIPP NA MARATONA DE PROGRAMAÇÃO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO

FRANCISCO VIRGINIO MARACCI
FRANCISCO ASSIS DA SILVA
LEANDRO LUIZ DE ALMEIDA
MÁRIO AUGUSTO PAZOTI
EMERSON DORIA
HAROLDO CESAR ALESSI
MOACIR DEL TREJO
AGLAÊ PEREIRA ZAUPA
CASSIA ALVES PEREGO
FLAVIO PANDUR ALBUQUERQUE CABRAL
HELTON MOLINA SAPIA

A Maratona de Programação é um evento da Sociedade Brasileira de Computação que existe desde o ano 2000. Ela nasceu das competições regionais classificatórias para as finais mundiais do concurso de programação da ACM, o ACM International Collegiate Programming Contest, e é parte da regional sul-americana do concurso. Ela se destina a alunos de cursos de graduação e início de pós-graduação na área de Computação. A competição promove nos alunos a criatividade, a capacidade de trabalho em equipe, a busca de novas soluções de software e a habilidade de resolver problemas sob pressão, e ainda dar continuidade no evento no qual a FIPP é uma das sedes da Olimpíada Brasileira de Informática. Várias universidades do Brasil desenvolvem concursos locais para escolher os melhores times para participar da Maratona de Programação. Estes times competem na Maratona (regional sul-americana) de onde os melhores serão selecionados para participar das Finais Nacionais e classificando-se, as mundiais do evento. A Faculdade de Informática iniciou a sua participação neste evento com alunos dos cursos de Ciência da Computação e Sistemas de Informação no ano de 2003. Desde 2004, o objetivo principal da FIPP é promover a participação dos alunos na maratona em incentivo a formação de uma estrutura base para a participação em eventos desta magnitude. Os alunos que participam da OBI acompanharam o treinamento e a participação dos times na Maratona de Programação da Sociedade Brasileira de Computação. Desde 2003 a FIPP participa com times compostos pelos alunos com melhores resultados na Maratona de Programação da FIPP. Estes times estão se apresentando em constante evolução com resultados expressivos se comparados aos resultados das competições anteriores, estimulando assim, a participação, treinamento e formação de novos times cada vez mais competitivos. Nos anos de 2007, 2009, 2012, 2013, 2015, 2016, 2017 e 2018 o time FIPP/UNOESTE, conquistou a 1ª colocação em uma das sedes regionais classificando se para a Final Nacional. A primeira fase da Maratona de Programação 2018 teve 746 times inscritos, de 219 diferentes escolas. A primeira fase ocorreu em 52 sedes espalhadas por todo o país. Além dos times inscritos, participaram 87 times "café com leite" totalizando 770 times competindo. A competição teve ainda a participação de times da Costa Rica, Nicaragua, Guatemala e México, num total de 1156 times simultâneos. Seis times brasileiros estarão presentes nas finais mundiais.

Ensino (ENAENS)

UNIVERSIDADE DO OESTE PAULISTA - UNOESTE

Ciências Exatas e da Terra

Comunicação oral

Ciência da Computação

15ª FESTA LINUX DA FACULDADE DE INFORMÁTICA DE PRESIDENTE PRUDENTE

ROBSON AUGUSTO SISCOOTTO
RONALDO TOSHIAKI OIKAWA
HAROLDO CESAR ALESSI
EMERSON DORIA
MOACIR DEL TREJO
HELTON MOLINA SAPIA
RENATO FERNANDO SILVA GONÇALVES
AGLAÊ PEREIRA ZAUPA
CASSIA ALVES PEREGO
FLAVIO PANDUR ALBUQUERQUE CABRAL
LEANDRO LUIZ DE ALMEIDA
FRANCISCO VIRGINIO MARACCI
SILVIO ANTONIO CARRO
FRANCISCO ASSIS DA SILVA
MÁRIO AUGUSTO PAZOTI

Linux é um sistema operacional alternativo ao Windows. Ele é estável, seguro e confiável. O Linux possui também uma grande seleção de programas para servidores, incluindo servidores Web, sistemas de banco de dados, linguagens de script e ferramentas para administração remota, servidores de e-mail, etc. Com o passar dos anos, o Linux vem ganhando mercado e com isso abrem-se mais oportunidades de emprego, novas tecnologias e maior economia por parte das empresas que trocam software proprietário por Software Livre e chegam a economizar milhões por ano com as licenças. Entretanto, a escassez de profissionais qualificados no mercado vem prejudicando esta evolução, o que justificativa a criação e um evento como a Festa Linux. A Festa Linux visa introduzir e proporcionar conhecimentos básicos aos estudantes principiantes no sistema operacional Linux. Além disso, por meio do Desafio Linux realizar uma competição, entre estudantes com conhecimentos básicos e avançados em Linux, visando a resolução de problemas referente à instalação e configuração de sistemas operacionais Linux e serviços de rede em tempo limitado. Estudos indicam que a busca por profissionais qualificados com habilidades no sistema operacional Linux é um item urgente no mercado de trabalho. A 15ª FESTA LINUX vai de encontro com essas necessidades qualificando profissionais. Órgão de fomento financiador da pesquisa: unoeste A 15ª Festa Linux foi realizada nas dependências da Faculdade de Informática de Presidente Prudente durante a 32ª Semana de Informática da FIPP/Unoeste. O evento teve uma duração de 32 horas, sendo 16 horas destinadas para o Linux para Principiantes e 16 horas para o Desafio Linux. Durante o Linux para principiantes, 100 alunos realizaram a instalação do sistema operacional Linux e tiveram a chance de atualizar, configurar e utilizar o sistema operacional. No 8º Desafio Linux, 10 equipes receberam uma lista de problemas que deveriam ser resolvidos, bem como seus respectivos pontos. Cada equipe submeteu seus problemas resolvidos a uma comissão julgadora que após avaliação atribuiu pontos ou uma penalidade. O time que resolveu o maior número de problemas corretos é declarado o vencedor. Os resultados obtidos nesta edição da Festa Linux são considerados satisfatórios e motivadores para as próximas edições, pois destacam-se o interesse da comunidade bem como a competição entre seus participantes. Diante disso foi possível desmistificar o Linux e colocar a prova conhecimentos necessários no mercado de trabalho.

Extensão (ENAEXT)

UNIVERSIDADE DO OESTE PAULISTA - UNOESTE

Ciências Exatas e da Terra

Comunicação oral

Ciência da Computação

15ª MARATONA DE PROGRAMAÇÃO DA FACULDADE DE INFORMÁTICA DE PRESIDENTE PRUDENTE
- UNOESTE

FRANCISCO VIRGINIO MARACCI
FRANCISCO ASSIS DA SILVA
EMERSON DORIA
HAROLDO CESAR ALESSI
LEANDRO LUIZ DE ALMEIDA
MÁRIO AUGUSTO PAZOTI
MOACIR DEL TREJO
ROBSON AUGUSTO SISCOOTTO
HELTON MOLINA SAPIA
AGLAÊ PEREIRA ZAUPA
CASSIA ALVES PEREGO

A Maratona de Programação da Faculdade de Informática de Presidente Prudente é um evento competitivo realizado anualmente nos laboratórios de informática de seus cursos. O torneio é composto por equipes de três alunos que recebem um caderno contendo problemas de níveis variados, que culminam no desafio de desenvolver algoritmos no menor tempo possível sem auxílio externo. O evento promove a integração dos alunos participantes, pois estimula a criatividade, habilidade, raciocínio, capacidade de resolver problemas sob pressão e trabalho em equipe. O fator competitivo deste torneio visa incentivar os alunos a participar de eventos de programação de nível nacional e internacional realizados anualmente, como a Olimpíada Brasileira de Informática, Maratona de Programação e o International Collegiate Programming Contest, entre outros. Em 2019 foram selecionados para a seletiva regional em Araçatuba, 12 alunos dos cursos de Ciência da Computação e Sistemas de Informação. Um dos resultados que merece destaque com a realização deste evento é a melhor preparação dos alunos para os torneios externos, que ocorrem posteriormente à realização deste, onde as melhores equipes são convidadas a participar. Este ano a maratona contou com a participação de 35 equipes compostas de 3 alunos cada, totalizando mais de 100 participantes, com o mesmo ambiente e regras das regionais realizadas pela Sociedade Brasileira de Computação no evento oficial nacional. A Maratona de Programação teve seu início no ano de 2005 com doze equipes, dos três cursos da Faculdade de Informática: Ciência da Computação, Sistemas de Informação e Tecnologias. Em 2012 o evento passou a contar com regulamento próprio de seleção de equipes para participação na Maratona de Programação da SBC com base nos resultados da Maratona da FIPP e em um cronograma de treinos desenvolvido para as equipes selecionadas. Eventos como este contribuem para demonstrar a importância do trabalho em equipe aos alunos, incentivo ao aprendizado de novas técnicas de desenvolvimento de software, aumento da motivação e interesse por disciplinas de algoritmos e programação, além de vislumbrar novos horizontes não explorados durante o decorrer da empreitada acadêmica.

Extensão (ENAEXT)

UNIVERSIDADE DO OESTE PAULISTA - UNOESTE

Ciências Exatas e da Terra

Comunicação oral

Ciência da Computação

2º HACKATHON FIPP - MARATONA DE ATIVIDADES TÉCNICAS DE COMPUTAÇÃO E INFORMÁTICA

AGLAÊ PEREIRA ZAUPA
ANA CAROLINA NICOLSI DA ROCHA GRACIOSO
SILVIO ANTONIO CARRO
SIDINEI DE OLIVEIRA SOUSA
CASSIA ALVES PEREGO
EMERSON DORIA
FLAVIO PANDUR ALBUQUERQUE CABRAL
FRANCISCO ASSIS DA SILVA
FRANCISCO VIRGINIO MARACCI
HAROLDO CESAR ALESSI
HELTON MOLINA SAPIA
LEANDRO LUIZ DE ALMEIDA
MÁRIO AUGUSTO PAZOTI
MOACIR DEL TREJO
ROBSON AUGUSTO SISCOOTTO

O Hackathon-FIPP é um evento que acontece durante a Semana de Computação e Informática da FIPP/Unoeste (INFOESTE) e reúne estudantes universitários dos cursos da Faculdade de Informática de Presidente Prudente (FIPP/Unoeste) para uma maratona de atividades na área de computação e informática, a partir de um tema pré-estabelecido. Tem por objetivo propor modelos, soluções e ideias inovadoras nas mais diversas áreas do conhecimento. Atualmente, no desenvolvimento das atividades propostas no programa de Estágio Supervisionado da FIPP/Unoeste identifica-se uma grande carência na qualidade e inovação do projeto de interfaces. Portanto, a temática escolhida para edição de 2019 foi interfaces de software, tendo como escopo uma concessionária que presta serviços de manutenção em veículos de uma marca autorizada. Diante desse cenário e considerando as boas práticas para a construção de interfaces existentes, as 10 equipes selecionadas desenvolveram o projeto das interfaces necessárias para executar o agendamento, registro de peças e serviços à Ordem de Serviço, e fechamento da conta do cliente dessa empresa. Com a realização do Hackathon-FIPP foram criados novos modelos de interface e os 3 melhores projetos servirão como referencial acadêmico para futuros projetos dos estudantes de Estágio Supervisionado da FIPP/Unoeste. O evento tem duração de dois dias. No primeiro dia, um profissional com notoriedade no mercado conduz o bootcamp que é um treinamento específico sobre o tema proposto. Já o segundo dia é dedicado à maratona de desenvolvimento na qual os estudantes têm a oportunidade de experimentar uma situação do mundo real, simulando problemas e desafios de um trabalho em grupo, onde o tempo e a concorrência são elementos presentes. Durante todo o período da maratona os mentores permanecem apoiando, esclarecendo dúvidas e acompanhando as equipes na melhoria das soluções concebidas. Esse formato permite que o estudante aprenda na prática em um ambiente descontraído, porém desafiador e inovador. Ao final do período da maratona as equipes são submetidas a uma avaliação realizada pela comissão julgadora que utiliza os critérios estabelecidos no início do evento. Para isso, cada equipe apresenta seu projeto em um pitch de 5 a 7 minutos que tem por objetivo despertar o interesse e promover o projeto, devendo conter apenas as informações essenciais e diferenciadas.

Extensão (ENAEXT)

UNIVERSIDADE DO OESTE PAULISTA - UNOESTE

Ciências Exatas e da Terra

Comunicação oral

Ciência da Computação

32ª SEMANA DE COMPUTAÇÃO E INFORMÁTICA DA FIPP/UNOESTE - INFOESTE 2019

FRANCISCO VIRGINIO MARACCI
AGLAÊ PEREIRA ZAUPA
CASSIA ALVES PEREGO
EMERSON DORIA
FRANCISCO ASSIS DA SILVA
HAROLDO CESAR ALESSI
HELTON MOLINA SAPIA
LEANDRO LUIZ DE ALMEIDA
MÁRIO AUGUSTO PAZOTI
MOACIR DEL TREJO
ROBSON AUGUSTO SISCOOTTO

A Faculdade de Informática de Presidente Prudente (FIPP) da Unoeste realiza, anualmente, a Semana de Computação e Informática da FIPP/Unoeste - INFOESTE. O evento, aberto a toda a comunidade regional e de outros centros, procura congrega estudantes de graduação e pós-graduação, professores, pesquisadores, profissionais e demais interessados em informática, objetivando a difusão da computação e informática e do seu bom uso, trazendo para o debate aberto temas importantes que indicam as tendências tecnológicas e do mercado atual. Uma das preocupações da FIPP é manter o seu corpo docente atualizado, visando dar o suporte necessário ao ensino das disciplinas oferecidas em seus cursos. É de interesse também, manter o seu corpo discente em contato com temas sempre atualizados visando um aprendizado maior, bem como a complementação do mesmo por parte de seus alunos. Promover cursos, visando à complementação e aprimoramento do conhecimento de docentes, discentes e demais profissionais da área. Promover palestras, workshops e oficinas visando à atualização de profissionais da área e alunos dos Cursos da FIPP. Possibilitar, aos alunos da Faculdade de Informática de Presidente Prudente e demais profissionais da área, a troca de experiências, bem como a atualização profissional através do conhecimento de novas tendências na área de informática e mercado de trabalho. Na edição de 2019 foram realizados 34 cursos (totalizando mais de 420 horas) e 5 palestras com duração de 3 horas cada (totalizando mais de 2300 vagas), além dos eventos integrados: 13ª Workshop sobre Educação em Computação e Informática da FIPP/UNOESTE (WEI-FIPP): duração de 3 horas, 15ª Festa Linux: duração de 24 horas; 15ª Maratona de Programação da FIPP: duração de 08 horas; 07 Ciclo de cursos de palestras para as ETEC'S (FIPPETEC): com aproximadamente 400 participantes; 26º Encontro de Alunos e Ex-alunos da FIPP/Unoeste: duração de 3 horas; e 2º Hackathon FIPP - Desafio de interfaces: duração média de 12 horas. Foram selecionados assuntos mais atuais para elaboração das palestras, workshops, oficinas e cursos. Os alunos da Faculdade de Informática e demais participantes tiveram a oportunidade de desfrutar de ótimos cursos que enfocaram temas atuais e tecnologias emergentes, além de excelentes palestras com palestrantes renomados que proporcionaram uma visão abrangente das novas tendências do mundo da Computação e Informática.

Extensão (ENAEXT)

UNIVERSIDADE DO OESTE PAULISTA - UNOESTE

Ciências Exatas e da Terra

Comunicação oral

Ciência da Computação

7º FIPPETEC - CICLO DE PALESTRAS E CURSOS DAS ETECS

CASSIA ALVES PEREGO
FRANCISCO VIRGINIO MARACCI
HAROLDO CESAR ALESSI
AGLAÊ PEREIRA ZAUPA
EMERSON DORIA
FLAVIO PANDUR ALBUQUERQUE CABRAL
FRANCISCO ASSIS DA SILVA
HELTON MOLINA SAPIA
LEANDRO LUIZ DE ALMEIDA
MÁRIO AUGUSTO PAZOTI
MOACIR DEL TREJO
ROBSON AUGUSTO SISCOOTTO
SILVIO ANTONIO CARRO

O FIPPETEC, Ciclo de Minicursos e Palestras, é um evento destinado às Escolas Técnicas da região de Presidente Prudente (ETECs), criado pela FIPP/Unoeste exclusivamente para alunos dos cursos técnicos das ETECs do Centro Paula Souza. O evento integra a programação da Semana de Computação e Informática (INFOESTE). Os cursos de informática oferecidos pelas ETECs frequentemente representam o primeiro contato dos alunos com a área de Computação e Tecnologia da Informação, muitos alunos prosseguem seus estudos nesta área. Dessa maneira, as ETECs constituem uma importante parceria para a promoção de projetos de extensão que visam levar a estes alunos possibilidades de formação e contato com uma instituição de nível superior. Tem como objetivo oferecer aos alunos das ETECs minicursos de temas atuais, auxiliando na formação técnica e também despertando o interesse para outras áreas da Computação e Informática. Os cursos abordaram assuntos atuais e relevantes para a área tecnológica, como: Modelagem 3D, HTML5 e CSS3, Construct 2, Introdução ao Photoshop, Desenho Digital, Realidade Virtual e Aumentada, Roteirização de Jogos e Introdução ao Arduíno. Os cursos são ministrados por profissionais da área externos, alunos dos últimos termos e docentes da instituição. Os minicursos foram muito bem avaliados pelos alunos apresentando em sua grande maioria uma atribuição de nota entre 8 a 10. Se os minicursos oferecidos no FIPPETEC tiveram o objetivo de auxiliar na formação do futuro profissional de TI e despertar o interesse por cursos superiores na área. No que se refere ao aspecto mais estrutural do evento, os alunos satisfeitos com a organização do FIPPETEC (nota superior ou igual a 5) é da ordem de 97%. É extremamente representativa a porcentagem (95%) dos alunos que consideram o FIPPETEC um elemento importante para sua formação profissional e, portanto, o objetivo do projeto em auxiliar na formação técnica e despertar o interesse pela formação superior na área foi atingido. Ao evento contou em sua sétima edição com 20 cursos de 4 horas de duração totalizando um total de 80 horas de treinamento. Foi oferecido um total de 400 vagas e 330 preenchidas. Os participantes são alunos de 7 ETECs, sendo convidadas apenas as escolas que ofereciam o curso de Técnico em Informática ou cursos correlatos (Adamantina, Dracena, Osvaldo Cruz, Presidente Prudente (duas ETECs), Presidente Venceslau e Teodoro Sampaio).

Ensino (ENAENS)

UNIVERSIDADE DO OESTE PAULISTA - UNOESTE

Comunicação oral

Ciências Exatas e da Terra

Ciência da Computação

PROGRAMANDO COM O SCRATCH: ENSINANDO LÓGICA DE PROGRAMAÇÃO E ALGORITMOS PARA ALUNOS INGRESSANTES DO ENSINO SUPERIOR.

CRISTIANE MACIEL RIZO
DANIELA TEREZA ASCENCIO RUSSI
CASSIA ALVES PEREGO
FRANCISCO ASSIS DA SILVA

O Scratch é um projeto do grupo Lifelong Kindergarten no Media Lab do MIT (Instituto de Tecnologia de Massachusetts), foi idealizado por Mitchel Resnick. Ele é fornecido gratuitamente online para os principais sistemas operacionais (Windows, Linux e Mac). A Faculdade de Informática de Presidente Prudente - FIPP conta com um Núcleo de Investigação e Práticas Pedagógicas - NIPP, que motivou o estudo por práticas diferenciadas e inovadoras para a readequação do conteúdo da disciplina Ambientes de Programação I. É comum estudantes de disciplinas de algoritmos e lógica de programação terem dificuldades de compreensão, assimilação e aprendizagem dos conteúdos no início dos seus estudos, por serem disciplinas complexas e desafiadoras. O relato de experiência tem por objetivo facilitar a introdução de conceitos de computação, bem como possibilitar o pensamento criativo e desenvolver o raciocínio lógico. Muitas universidades estão buscando maneiras de tornar a aprendizagem de algoritmos interessante, divertida e produtiva. O Scratch permite ao estudante criar animações, histórias e games, de maneira simples e interativa. A nova proposta metodológica adotada foi bem interessante, pois gamificou o conteúdo trabalhado na disciplina. Como resultado cada grupo de estudantes desenvolveu um jogo como projeto final do bimestre. Ficou evidente que o uso do Scratch contribuiu para motivar e melhorar o desempenho dos estudantes nas disciplinas de algoritmos e lógica de programação. A partir dessas observações, pode-se concluir que o ensino de lógica de programação, com o uso do Scratch, possibilita buscar alternativas para apoiar a aprendizagem de algoritmos significativa e divertida. Entre os recursos do Scratch, destacam-se as competências para a resolução de problemas, para a concepção de projetos com raciocínio lógico, decomposição de problemas complexos em partes mais simples, identificação e eliminação de erros, desenvolvimento de ideias, desde a concepção até a concretização do projeto, estimulando a concentração e a perseverança. O ambiente define áreas onde o programador deve escolher as ferramentas que vai utilizar: palco, onde é visualizado o conteúdo produzido; atores, onde ficam dispostos os atores que podem aparecer no palco; e uma área dividida em três categorias de comandos: código, onde ficam todos os blocos de instruções disponíveis; trajes, onde o estudante pode editar as fantasias dos personagens; sons, onde o aluno pode editar ou criar um novo som.

Extensão (ENAEXT)

UNIVERSIDADE DO OESTE PAULISTA - UNOESTE

Ciências Exatas e da Terra

Comunicação oral

Ciência da Computação

PROGRAMANDO O FUTURO

FRANCISCO VIRGINIO MARACCI
DANIELA TEREZA ASCENCIO RUSSI
EMERSON DORIA
HAROLDO CESAR ALESSI
MOACIR DEL TREJO
AGLAÊ PEREIRA ZAUPA
CASSIA ALVES PEREGO
FRANCISCO ASSIS DA SILVA
HELTON MOLINA SAPIA
LEANDRO LUIZ DE ALMEIDA
MÁRIO AUGUSTO PAZOTI

Segundo a Sociedade Brasileira de Computação (SBC), no mundo inteiro o número de alunos interessados na área de informática vem diminuindo significativamente devido à dificuldade da introdução à informática com ênfase em programação, principalmente, pelos problemas advindos do sistema de ensino básico. Com o intuito de promover e estimular o interesse pela área, a SBC realiza a Olimpíada Brasileira de Informática (OBI), com participação de instituições de ensino fundamental e médio de forma gratuita. Nesta perspectiva, o objetivo do projeto é despertar o interesse pela área de informática em alunos do ensino fundamental e médio por intermédio de capacitações com vistas à participação em competições como a OBI. Para isso é necessário introduzir e desenvolver lógica de programação em capacitações divididas em dois níveis. A primeira turma capacitada obteve duas (2) medalhas de bronze e uma (1) de prata. A segunda turma obteve nove (9) classificados para segunda fase da OBI dentro de dez (10) participantes e ótimas colocações na segunda fase a nível nacional chegando aos dois por cento (2%) melhores nas classificações. A terceira turma obteve uma (1) medalha de ouro, duas (2) medalha de prata e duas (2) medalha de bronze. A quarta obteve ótimas classificações a nível nacional mantendo-se entre os cinco (5%) por cento dos melhores a nível Nacional. A quinta turma conta com nove (9) participantes para a terceira fase (final nacional) que irá ocorrer em doze de agosto de 2017. A sétima turma obteve uma (1) medalha de prata e alunos mantendo-se entre os cinco (5%) por cento dos melhores a nível Nacional. A oitava turma conta com 15 aprovados para a segunda fase da competição que irá ocorrer neste ano de 2019. Os aprovados na segunda fase irão para a terceira fase (final nacional) momento em que é definida as classificações. Atualmente, o projeto encerra a oitava edição com a participação dos alunos na primeira (local), segunda (regional) e terceira fase da OBI (nacional) da OBI e inicia a seleção de alunos para a nona edição de capacitação com duração de um ano. O projeto, em sua oitava edição, contou com alunos do 6º e 8º anos de cinco colégios particulares de Presidente Prudente sendo uma turma de cada nível, da modalidade iniciação, tendo ao todo 50 (cinquenta) alunos participantes. Durante as capacitações é desenvolvido a lógica de programação por intermédio de jogos educativos, resolução de provas anteriores da competição e desafios.

Ensino (ENAENS)

UNIVERSIDADE DO OESTE PAULISTA - UNOESTE

Comunicação oral

Ciências Exatas e da Terra

Ciência da Computação

RECURSOS COMPUTACIONAIS PARA APOIO NO ENSINO DA DISCIPLINA DE ALGORITMO E LÓGICA DE PROGRAMAÇÃO

MARCOS REIS FERREIRA
GIOVANA ANGÉLICA ROS MIOLA

O acesso às Tecnologias da Informação e da Comunicação, está disponível desde muito cedo. Os jovens que hoje estão iniciando seus estudos no nível superior já têm contato com tais tecnologias possuindo acesso a quantidades enormes de informação nos mais diversos formatos. A motivação em desenvolver este trabalho se deu durante o segundo semestre de 2018 e o primeiro semestre de 2019, momento em que foi realizada a monitoria nas disciplinas de Algoritmo, Linguagem de Programação e Estrutura de Dados do curso de Análise e Desenvolvimento de Sistemas da Fatec de Presidente Prudente. Durante este período, em contato direto com os alunos, foi observada, as principais dificuldades no processo de aprendizagem do conteúdo, bem como os obstáculos que frequentemente acomete os alunos. Os objetivos deste estudo, consistiu em: investigar as dificuldades mais comumente encontradas pelos alunos, apresentar ferramentas pedagógicas e tecnológicas para tratar tais dificuldades e propor um roteiro de aula para a disciplina de Algoritmo e Lógica de Programação que antecipasse estas dificuldades tratando-as antes que de fato surgissem. No geral, as ferramentas utilizadas impactaram positivamente na dinâmica da aula, comprovados pela demonstração de relatórios de desempenho dos alunos. O Google Sala de Aula se destacou por sua facilidade de gerenciamento e compartilhamento de arquivos e atividades com os alunos. A plataforma Code.org se mostrou eficiente em antecipar ao aluno, de uma maneira visual, os conceitos que seriam trabalhados no decorrer do semestre. A linguagem de programação Python foi efetiva em seu propósito de facilitar a escrita de algoritmos pelos alunos, sua sintaxe mais simples e próxima da linguagem natural reduziu o esforço do aluno em entender em dominar a escrita da linguagem. Tais dados, além de serem úteis durante a disciplina, podem ser acumulados e estudados no decorrer de várias turmas o que permite ao professor avaliar continuamente suas práticas e observar concretamente seus efeitos, colaborando com um mecanismo de aperfeiçoamento contínuo. Foram utilizadas as ferramentas: Google Sala de aula, Code.org e Google Colab, como mecanismos para auxílio no processo de aprendizado, mediando troca de arquivos e mensagens entre professores e alunos, organizando os dados gerados, oferecendo ambiente de programação, utilizando a linguagem de programação Python em navegadores, possibilitando facilmente ser acessados pelo professor e alunos.