



UNIVERSIDADE FEDERAL
DE ALAGOAS

INSTITUTO DE COMPUTAÇÃO PROCESSAMENTO DIGITAL DE IMAGENS

RELATÓRIO REPLICAÇÃO DO PROJETO ORIGINAL

Nome: Lucas Mendes Massa

Matrícula: 18112211

Nome: Jhonnye Gabriel de Oliveira Farias

Matrícula: 18112203

REPOSITÓRIO DO TRABALHO ORIGINAL.

O presente trabalho teve como objetivo abordar a força que os métodos clássicos de processamento de imagem, quando bem estruturados, conseguem superar os algoritmos direcionados a dados, como redes neurais. O paper aborda um algoritmo simples e fácil que roda na CPU, sua outra principal vantagem é que por não necessitar de treinamento, o mesmo poupa recursos em quesito de memória e na busca por base de dados. O algoritmo foi avaliado seu *benchmarking* na base de dados de preenchimento de profundidade KITTI, que na data de sua submissão atingiu o rank N° 1.

O preenchimento de profundidade é uma técnica que deriva da tarefa de converter mapas de profundidade esparsos em mapas de profundidade densa. Essa técnica foi inicialmente desenvolvida para ajudar na visualização de resultados de detecção de objetos 3D para o [AVOD](#). Um algoritmo preciso de preenchimento de profundidade pode ajudar não somente na visualização dos objetos mas também em algoritmos SLAM que utilizam entradas de nuvens de pontos.

O método empregado toma proveito de projeções do lidar, não havendo então a necessidade de utilizar as imagens.

BASE DE DADOS UTILIZADA

O algoritmo foi avaliado na base de dados de benchmarking de preenchimento de profundidade KITTI, que não só avalia como permite treinar modelos de deep learning complexos para as tarefas de preenchimento de profundidade e predição de profundidade de imagem única. As métricas empregadas na avaliação foram: iRMSE, RMSE, iMAE, MAE.

REPLICAÇÃO E VALIDAÇÃO

Primeiramente tratando dos dados aplicados nos experimentos, como foi mencionado anteriormente, o projeto faz o uso da base de dados KITTI, mais especificamente seu módulo de Depth Completion. Dentre as pastas de arquivos presentes no mesmo, foi utilizada a de nome “val_selection_cropped”. Essa pasta, por sua vez, é dividida entre três conteúdos principais: `groundtruth_depth` (contempla os mapas de densidade que se deseja atingir), `image` (reúne as imagens originais correspondentes aos mapas de densidade) e `velodyne_raw` (pasta formada pelo conjunto de mapas de densidade esparsos usados como entrada).

No que se refere ao código utilizado, o mesmo pode ser encontrado no repositório original dos autores. O projeto é implementado fazendo uso somente da linguagem python e está dividido entre três arquivos principais: `depth_completion.py` (arquivo que realiza a leitura das imagens e chama o algoritmo), `depth_map_utils.py` (arquivo no qual o algoritmo de depth completion está implementado) e `vis_utils.py` (possui algumas funções de visualização implementadas). Com relação às bibliotecas necessárias para a execução do projeto (excluindo as padrão), apenas quatro estão presentes no arquivo de requisitos: `numpy`, `open-cv`, `matplotlib` e `pypng`.

A implementação do algoritmo foi realizada através da ferramenta Jupyter Notebook, tendo em vista que a mesma permite uma melhor visualização de resultados instantâneos. Primeiramente foi feita uma replicação do código original do projeto. Em seguida, o algoritmo foi executado em dois mapas de densidade escolhidos da base de dados a fim de reproduzir os resultados vistos no artigo. Por fim, foi feita uma validação do algoritmo a partir da extração de métricas com relação a todas os mapas de densidade presentes no módulo mencionado acima, mais especificamente Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) e o tempo médio de execução do algoritmo. Os detalhes a respeito da implementação podem ser vistos no Jupyter Notebook presente no repositório.

Os resultados se mostraram muito similares aos do artigo original, o que valida a replicação do projeto. No que concerne às dificuldades, não foram encontradas muitas para a reprodução do algoritmo. Inicialmente não estava claro quais dados foram usados para medir a performance do algoritmo, tendo em vista que a base de depth completion é formada por várias subpastas. Analisando o código do projeto foi possível identificar de qual pasta os mapas de densidade eram lidos. Também houve dificuldades quanto à qual era a configuração do algoritmo que teve melhor desempenho, o que não estava claro no artigo. Analisando o repositório e o código foi possível identificar a configuração. Por fim, vale citar que os nomes dos arquivos da base se mostraram um pouco confusos de se entender.