Im2col

Simple Usage

See `main.cpp`.

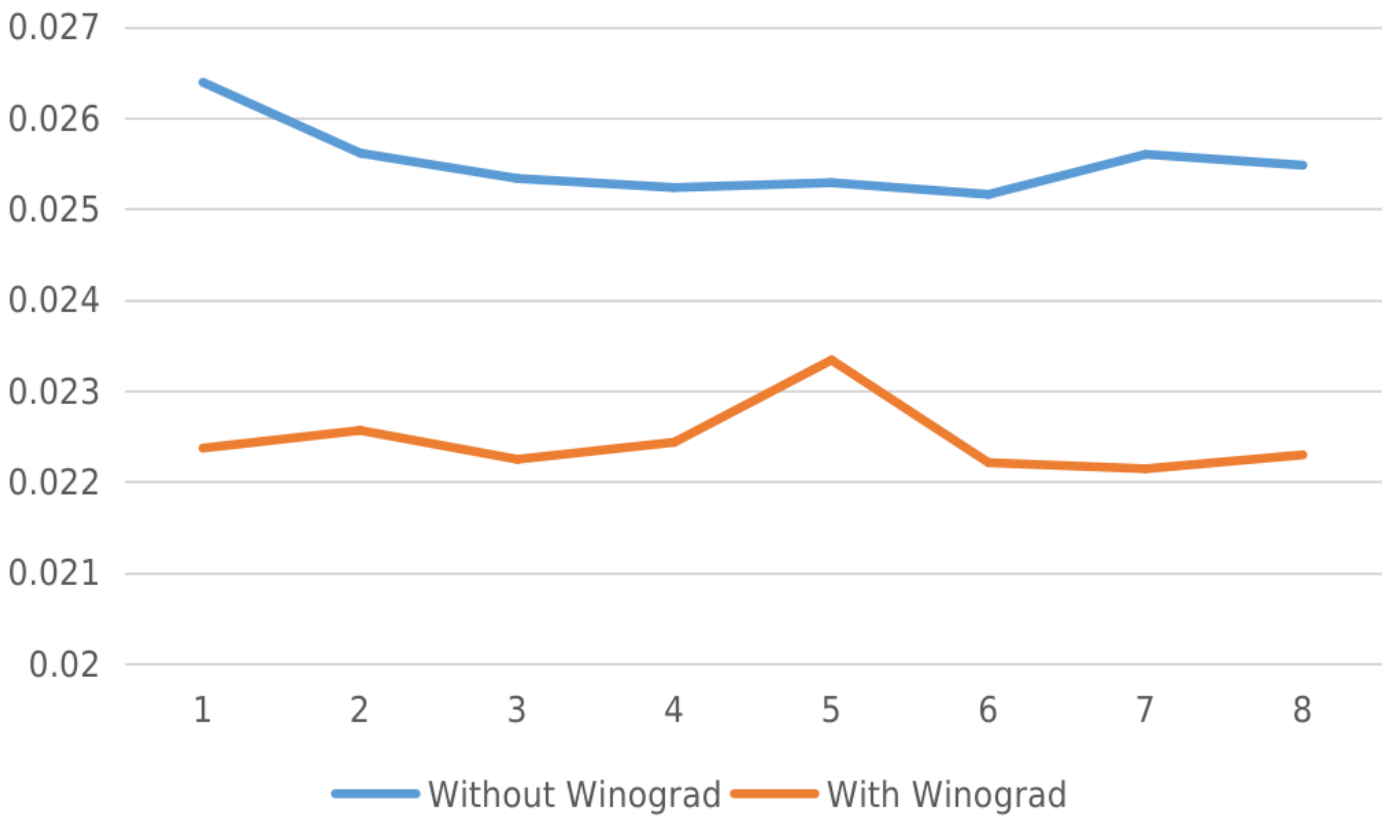
Files and Directories

```
Im2col
├── acc_function
│   ├── AcceleratorFunction.h        // The base class of accelertor functions.
│   └── winograd
│       ├── WinogradFunction_1D.cpp  // Winograd 1D. (****** HERE!!!!!! ******)
│       ├── WinogradFunction_1D.h
│       └── WinogradFunction.h       // The base classWinograd accelertor funct
├── CMakeLists.txt
├── feature_map
│   ├── DirectFeatureMap.cpp
│   ├── DirectFeatureMap.h
│   ├── FeatureMap.h
│   ├── Im2colFeatureMap.cpp         // Im2col feature map that allows
│   │                                   convolution with a accelerator
│   │                                   function. (****** HERE!!!!!! ******)
│   ├── Im2colFeatureMap.h
│   ├── Im2colFeatureMap_OMP.cpp
│   └── Im2colFeatureMap_OMP.h
├── kernel
│   ├── DirectKernel.cpp
│   ├── DirectKernel.h
│   ├── Im2colKernel.cpp
│   ├── Im2colKernel.h
│   └── Kernel.h
├── main.cpp
├── output_map
│   ├── OutputMap.cpp
│   └── OutputMap.h
└── util
    └── GetTime.h
```

Analysis

```
Direct conv: 0.03022
Im2col conv: 0.0263898
Im2col conv with Winograd: 0.0223682
Im2col conv with Winograd (OpenMP enabled): 0.00318503
```

Steady Improvement is Witnessed when Using Winograd



In Lab2 settings, my implementation is around 13% faster on average when using Winograd(2, 3), which is not beyond expectation because, in fact, Winograd does not break much spatial locality. As the size of a row is small under our settings, although it loads two rows at the same time, we can infer that it still needs not to reload data from memory to cache.

Parallelism

Still, we can use OpenMP to parallel convolution even if Winograd is used. Yet, this time we cannot simply add the pragma in the innermost for-loop because the innermost for-loop will not loop many times when we use Winograd (e.g., loop for K * R * S / 3 times when using Winograd(2, 3)). Here we'd better let one thread be responsible for every two rows, which guarantees little parallel overhead (and also better locality actually).

$$\left[\begin{array}{ccc|ccc|ccc} d_{00} & d_{01} & d_{02} & d_{10} & d_{11} & d_{12} & d_{20} & d_{21} & d_{22} \\ d_{01} & d_{02} & d_{03} & d_{11} & d_{12} & d_{13} & d_{21} & d_{22} & d_{23} \\ \hline d_{10} & d_{11} & d_{12} & d_{20} & d_{21} & d_{22} & d_{30} & d_{31} & d_{32} \\ d_{11} & d_{12} & d_{13} & d_{21} & d_{22} & d_{23} & d_{31} & d_{32} & d_{33} \end{array}\right] \left[\begin{array}{c} k_{00} \\ k_{01} \\ k_{02} \\ \hline k_{10} \\ k_{11} \\ k_{12} \\ \hline k_{20} \\ k_{21} \\ k_{22} \end{array}\right] = \left[\begin{array}{c} r_{00} \\ r_{01} \\ \hline r_{10} \\ r_{11} \end{array}\right]$$