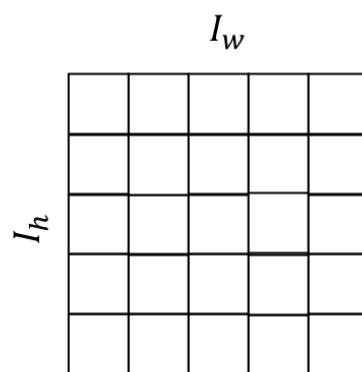


## 7.2 im2colとcol2im

- 畳み込み層とプーリング層のコードをシンプルに保ち、実行速度を早く保つために、im2colおよびcol2imというアルゴリズムを用いる。
- im2colは、画像を表す4次元配列を行列に変換し、col2imは行列を、画像を表す4次元配列に変換する。

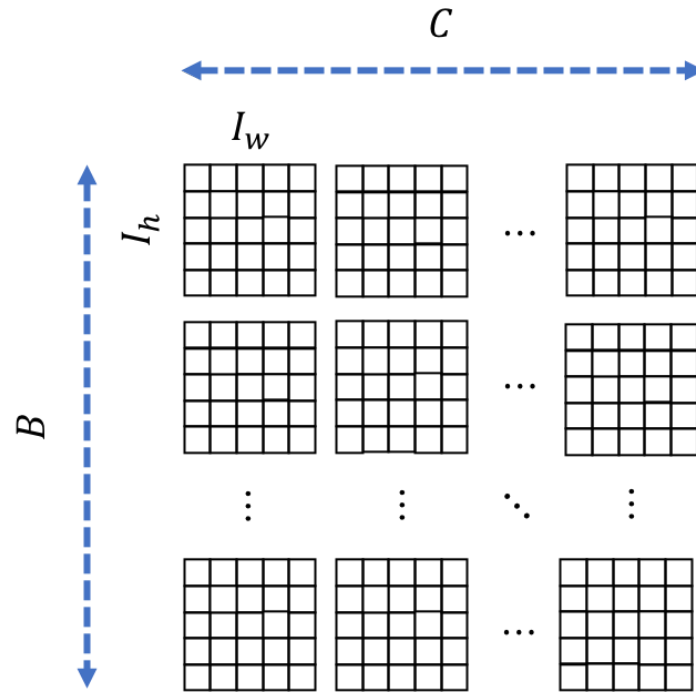
### 7.2.1 im2colとcol2imの概要

- 畳み込み層への入力は、バッチとチャンネルを考慮しなければ、次のようなシンプルな行列で表すことができる：



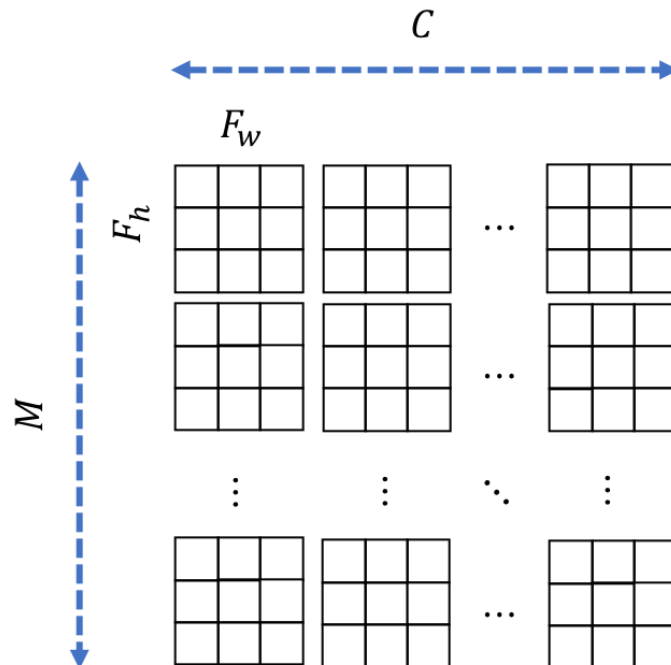
#### □畳み込み層への入力画像

- しかしながら、実際にはバッチ学習、ミニバッチ学習に対応する必要があり、複数のチャンネルを持つRGB画像を扱う必要もある。
- バッチ、チャンネルに対応した入力画像を次に示す：



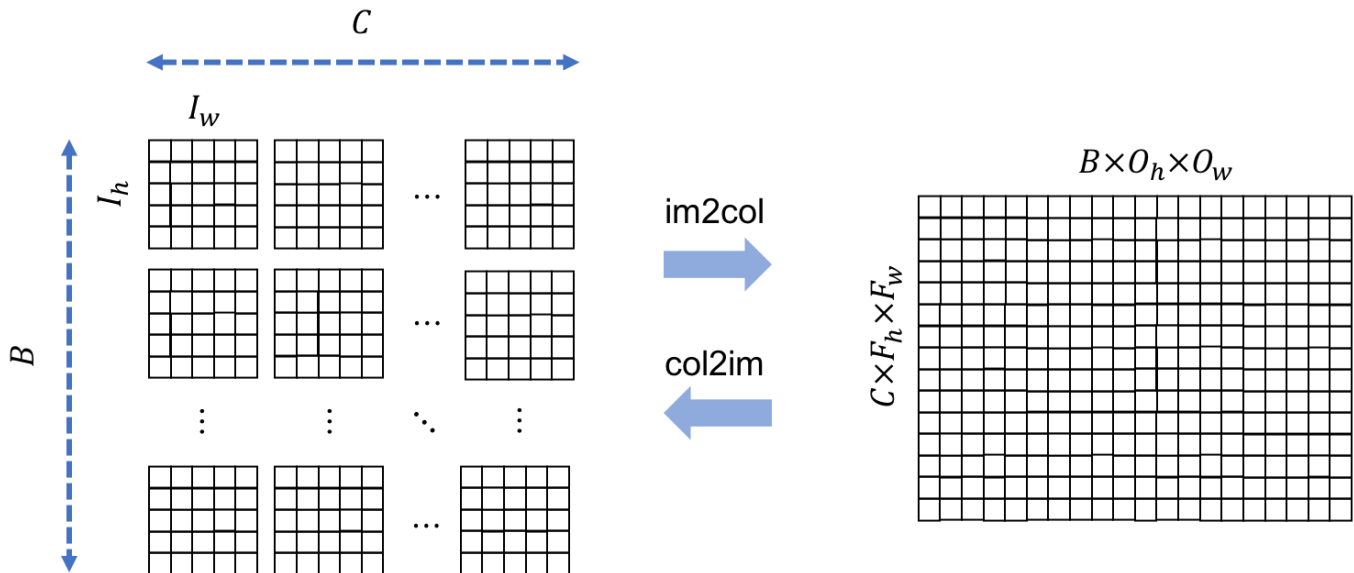
#### □ バッチとチャンネル数を考慮した場合の畳み込み層への入力画像

- 上図に示すように、入力画像は4次元配列、すなわち4階のテンソルになる。
- そして、フィルタは複数あり、それぞれのフィルタのチャンネル数は入力画像のチャンネル数と同じになる：



#### □ フィルタの全体像

- 以上のように、畳み込み層を実装する際には多くの入り組んだ要素を考慮する必要があり、何重にも重なった多次元配列を扱う必要がある。
- 多次元配列をfor文などによるループで扱おうとすると、ループが幾重にもなりコードが複雑になってしまう。
- また、NumPyは、行列演算は高速だが、ループの実行にはとても時間がかかる。
- そこで、ループを最小限に抑え、コードをシンプルに保つために用いるのが、im2colおよびcol2im：



#### □im2colとcol2imによる変換

- 順伝播の場合は`im2col`、逆伝播の場合は`col2im`が用いられるが、これらのアルゴリズムにより実行時間のかかるループ処理を最小限に抑え、メインの計算を1つの行列積に集約させることが可能になる。
- `im2col`と`col2im`は、プーリング層でも活用される。

## 7.2.2 im2colのアルゴリズム

•

## 7.2.3 im2colの実装 -シンプルなim2col-