We discuss the generation of $\phi(\text{pad}(Z^{m,i}))$

`im2col` in Existing Packages I

- Due to the wide use of CNN, a subroutine for $\phi(\text{pad}(Z^{m,i}))$ has been available in some packages
- For example, MATLAB has a built-in function `im2col` that can generate $\phi(\text{pad}(Z^{m,i}))$ for

$$s = 1 \text{ and } s = h \text{ (width of filter)}$$

- But this function cannot handle general s
- Can we do a reasonably efficient implementation by ourselves?

`im2col` in Existing Packages II

- For an easy description we consider

$$\text{pad}(Z^{m,i}) = Z^{\text{in},i} \quad \rightarrow \quad Z^{\text{out},i} = \phi(Z^{\text{in},i}).$$

Linear Indices and an Example I

- For the matrix

$$
Z^{\text{in},i} = \begin{bmatrix}
z^i_{1,1,1} & z^i_{2,1,1} & \cdots & z^i_{a^{\text{in}},1,1} & z^i_{1,2,1} & \cdots & z^i_{a^{\text{in}},b^{\text{in}},1} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots \vdots & \\
z^i_{1,1,j} & z^i_{2,1,j} & \cdots & z^i_{a^{\text{in}},1,j} & z^i_{1,2,j} & \cdots & z^i_{a^{\text{in}},b^{\text{in}},j} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots \vdots & \\
z^i_{1,1,d^{\text{in}}} & z^i_{2,1,d^{\text{in}}} & \cdots & z^i_{a^{\text{in}},1,d^{\text{in}}} & z^i_{1,2,d^{\text{in}}} & \cdots & z^i_{a^{\text{in}},b^{\text{in}},d^{\text{in}}}
\end{bmatrix}
$$

we count elements in a column-oriented way

Linear Indices and an Example II

- This leads to the following column-oriented linear indices of $Z^{\text{in},i}$:

$$\begin{bmatrix} 1 & d^{\text{in}}+1 & \dots & (b^{\text{in}}a^{\text{in}}-1)d^{\text{in}}+1 \\ 2 & d^{\text{in}}+2 & \dots & (b^{\text{in}}a^{\text{in}}-1)d^{\text{in}}+2 \\ \vdots & \vdots & \ddots & \vdots \\ d^{\text{in}} & 2d^{\text{in}} & \dots & (b^{\text{in}}a^{\text{in}})d^{\text{in}} \end{bmatrix} \in R^{d^{\text{in}} \times a^{\text{in}} b^{\text{in}}}. \tag{1}$$

- We know every element in

$$\phi(Z^{\text{in},i}) \in R^{hhd^{\text{in}} \times a^{\text{out}} b^{\text{out}}},$$

is extracted from $Z^{\text{in},i}$

Linear Indices and an Example III

- Thus the task is to find the mapping between each element in $\phi(Z^{\text{in},i})$ and a linear index of $Z^{\text{in},i}$.
- Consider an example with

$$a^{\text{in}} = 3, \; b^{\text{in}} = 2, \; d^{\text{in}} = 1.$$

  Because $d^{\text{in}} = 1$, we omit the channel subscript.
- In addition, we omit the instance index i, so the image is

$$\begin{bmatrix} z_{11} & z_{12} \\ z_{21} & z_{22} \\ z_{31} & z_{32} \end{bmatrix}. \tag{2}$$

Linear Indices and an Example IV

- If

$$h = 2, s = 1,$$

two sub-images are

$$\begin{bmatrix} z_{11} & z_{12} \\ z_{21} & z_{22} \end{bmatrix} \text{ and } \begin{bmatrix} z_{21} & z_{22} \\ z_{31} & z_{32} \end{bmatrix}$$

- By our earlier way of representing images,

$$Z^{\text{in},i} = \begin{bmatrix} z^i_{1,1,1} & z^i_{2,1,1} & \cdots & z^i_{a^{\text{in}},b^{\text{in}},1} \\ \vdots & \vdots & \ddots & \vdots \\ z^i_{1,1,d^{\text{in}}} & z^i_{2,1,d^{\text{in}}} & \cdots & z^i_{a^{\text{in}},b^{\text{in}},d^{\text{in}}} \end{bmatrix}$$

Linear Indices and an Example V

we now have

$$Z^{\text{in}} = \begin{bmatrix} z_{11} & z_{21} & z_{31} & z_{12} & z_{22} & z_{32} \end{bmatrix}$$

- The linear indices from (1) are

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \end{bmatrix}.$$

Linear Indices and an Example VI

- Recall that

$$\phi(Z^{\text{in},i}) =$$

$$\begin{bmatrix} z^i_{1,1,1} & z^i_{1+s,1,1} & & z^i_{1+(a^{\text{out}}-1)s,1+(b^{\text{out}}-1)s,1} \\ z^i_{2,1,1} & z^i_{2+s,1,1} & & z^i_{2+(a^{\text{out}}-1)s,1+(b^{\text{out}}-1)s,1} \\ \vdots & \vdots & \ldots & \vdots \\ z^i_{h,h,1} & z^i_{h+s,h,1} & & z^i_{h+(a^{\text{out}}-1)s,h+(b^{\text{out}}-1)s,1} \\ \vdots & \vdots & & \vdots \\ z^i_{h,h,d^{\text{in}}} & z^i_{h+s,h,d^{\text{in}}} & & z^i_{h+(a^{\text{out}}-1)s,h+(b^{\text{out}}-1)s,d^{\text{in}}} \end{bmatrix}$$

Linear Indices and an Example VII

- Therefore,

$$\phi(Z^{\text{in}}) = \begin{bmatrix} z_{11} & z_{21} \\ z_{21} & z_{31} \\ z_{12} & z_{22} \\ z_{22} & z_{32} \end{bmatrix}. \qquad (3)$$

- Let's check linear indices using Matlab/Octave

```
octave:8> reshape((1:6)', 3, 2)
ans =

   1   4
   2   5
```

Linear Indices and an Example VIII

```
    3    6
```

This gives linear indices of the image in (2)

```
octave:9> im2col(reshape((1:6)', 3, 2),
                     [2,2], "sliding")
ans =

    1    2
    2    3
    4    5
    5    6
```

Here [2 2] is the filter size.

Linear Indices and an Example IX

- Clearly,

  1
  2
  4
  5

  correspond to linear indices of the first column in $\phi(Z^{\text{in}})$; see (3)

Linear Indices and an Example X

- To handle all instances together, we store

$$Z^{\text{in},1}, \ldots, Z^{\text{in},l}$$

as

$$\begin{bmatrix} \text{vec}(Z^{\text{in},1}) & \ldots & \text{vec}(Z^{\text{in},l}) \end{bmatrix} \tag{4}$$

- For our example,

$$\text{vec}(Z^{m,i}) = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{bmatrix}, \quad \text{vec}(\phi(Z^{m,i})) = \begin{bmatrix} 1 \\ 2 \\ 4 \\ 5 \\ 2 \\ 3 \end{bmatrix}$$

Linear Indices and an Example XI

- Denote (4) as a MATLAB matrix

$$\texttt{Z}$$

- Then

$$\begin{bmatrix} \text{vec}(\phi(Z^{m,1})) & \dots & \text{vec}(\phi(Z^{m,l})) \end{bmatrix}$$

  is simply

$$\texttt{Z(P,:)}$$

  in MATLAB, where we store the mapping by

$$\texttt{P} = [\, 1 \;\; 2 \;\; 4 \;\; 5 \;\; 2 \;\; 3 \;\; 5 \;\; 6 \,]^T$$

Linear Indices and an Example XII

- All instances are handled in one line
- Moreover, we hope that Matlab's implementation on this mapping operation is efficient
- But how to obtain `P`?
- Note that

$$\begin{bmatrix}1 & 2 & 4 & 5 & 2 & 3 & 5 & 6\end{bmatrix}^T.$$

also corresponds to column indices of non-zero elements in P_ϕ^m.

Linear Indices and an Example XIII

$$
\begin{bmatrix} z_{11} \\ z_{21} \\ z_{12} \\ z_{22} \\ z_{21} \\ z_{31} \\ z_{22} \\ z_{32} \end{bmatrix}
=
\begin{bmatrix}
1 & & & & & \\
 & 1 & & & & \\
 & & & 1 & & \\
 & & & & 1 & \\
 & 1 & & & & \\
 & & 1 & & & \\
 & & & & 1 & \\
 & & & & & 1
\end{bmatrix}
\begin{bmatrix} z_{11} \\ z_{21} \\ z_{31} \\ z_{12} \\ z_{22} \\ z_{32} \end{bmatrix}
\quad (5)
$$

- Each row has a single non-zero
- The column index of the non-zero indicates the extraction of elements from Z^{in} to $\phi(Z^{\text{in}})$

Finding the Mapping I

- We begin with checking how linear indices of $Z^{\text{in},i}$ can be mapped to the first column of $\phi(Z^{\text{in},i})$.
- For simplicity, we consider channel j first
- From

$$Z^{\text{in},i} = \begin{bmatrix} z^i_{1,1,1} & z^i_{2,1,1} & \cdots & z^i_{a^{\text{in}},1,1} & z^i_{1,2,1} & \cdots & z^i_{a^{\text{in}},b^{\text{in}},1} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ z^i_{1,1,j} & z^i_{2,1,j} & \cdots & z^i_{a^{\text{in}},1,j} & z^i_{1,2,j} & \cdots & z^i_{a^{\text{in}},b^{\text{in}},j} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ z^i_{1,1,d^{\text{in}}} & z^i_{2,1,d^{\text{in}}} & \cdots & z^i_{a^{\text{in}},1,d^{\text{in}}} & z^i_{1,2,d^{\text{in}}} & \cdots & z^i_{a^{\text{in}},b^{\text{in}},d^{\text{in}}} \end{bmatrix}$$

Finding the Mapping II

for the first column of $\phi(Z^{\text{in},i})$, for channel j, we select an $h \times h$ matrix from $Z^{\text{in},i}$.

Finding the Mapping III

- The selected values and their linear indices are

values	linear indices in $Z^{\text{in},i}$
$z_{1,1,j}$	j
$z_{2,1,j}$	$d^{\text{in}} + j$
$\vdots$	$\vdots$
$z_{h,1,j}$	$(h-1)d^{\text{in}} + j$
$z_{1,2,j}$	$a^{\text{in}}d^{\text{in}} + j$
$\vdots$	$\vdots$
$z_{h,2,j}$	$((h-1) + a^{\text{in}})d^{\text{in}} + j$
$\vdots$	$\vdots$
$z_{h,h,j}$	$((h-1) + (h-1)a^{\text{in}})d^{\text{in}} + j$

Finding the Mapping IV

- We take d^{in} out and rewrite the earlier table as

$$\begin{bmatrix} 0 + 0a^{\text{in}} \\ \vdots \\ (h-1) + 0a^{\text{in}} \\ 0 + 1a^{\text{in}} \\ \vdots \\ (h-1) + 1a^{\text{in}} \\ \vdots \\ 0 + (h-1)a^{\text{in}} \\ \vdots \\ (h-1) + (h-1)a^{\text{in}} \end{bmatrix} d^{\text{in}} + j = \begin{bmatrix} 0 \\ \vdots \\ h-1 \\ 0 \\ \vdots \\ h-1 \\ \vdots \\ 0 \\ \vdots \\ h-1 \end{bmatrix} d^{\text{in}} + j + \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ \vdots \\ 1 \\ \vdots \\ h-1 \\ \vdots \\ h-1 \end{bmatrix} a^{\text{in}} d^{\text{in}} \quad (6)$$

Finding the Mapping V

- We will see that the right-side of (6) will be used in the practical implementation
- Every linear index in (6) can be represented as

$$(p + qa^{\text{in}})d^{\text{in}} + j, \tag{7}$$

  where

$$p,\ \ q \in \{0, \ldots, h-1\}$$

- Thus $(p+1, q+1)$ corresponds to the pixel position in the first sub-image (or say the convolutional filter)

Finding the Mapping VI

- Next we consider other columns in $\phi(Z^{\text{in},i})$ by still fixing the channel to be j.

Finding the Mapping VII

- From

$$\phi(Z^{\text{in},i}) =$$

$$\begin{bmatrix} \vdots & \vdots & & \vdots \\ z^i_{1,1,j} & z^i_{1+s,1,j} & & z^i_{1+(a^{\text{out}}-1)s,1+(b^{\text{out}}-1)s,j} \\ z^i_{2,1,j} & z^i_{2+s,1,j} & & z^i_{2+(a^{\text{out}}-1)s,1+(b^{\text{out}}-1)s,j} \\ \vdots & \vdots & \dots & \vdots \\ z^i_{h,h,j} & z^i_{h+s,h,j} & & z^i_{h+(a^{\text{out}}-1)s,h+(b^{\text{out}}-1)s,j} \\ \vdots & \vdots & & \vdots \end{bmatrix}$$

Finding the Mapping VIII

each column contains the following elements from the jth channel of $Z^{\text{in},i}$.

$$z_{1+p+as,1+q+bs,j}\ , \quad \begin{aligned} & a = 0,1,\ldots,a^{\text{out}}-1, \\ & b = 0,1,\ldots,b^{\text{out}}-1, \\ & p,\ q \in \{0,\ldots,h-1\}. \end{aligned} \tag{8}$$

Clearly, when

$$p = q = 0,$$

we have that

$$(1+as,\ 1+bs)$$

Finding the Mapping IX

is the top-left position of a sub-image in the channel j of $Z^{\text{in},i}$.

- This is reasonable as we now use stride s to generate $a^{\text{out}} b^{\text{out}}$ sub-images
- The linear index of each element in (8) is

$$
\begin{aligned}
&\underbrace{((1+p+as-1)+(1+q+bs-1)a^{\text{in}})}_{\text{column index in } Z^{\text{in},i} - 1} d^{\text{in}} + j \\
=&((p+as)+(q+bs)a^{\text{in}})d^{\text{in}} + j \\
=& (a+ba^{\text{in}})sd^{\text{in}} + \underbrace{(p+qa^{\text{in}})d^{\text{in}} + j}_{\text{see (7)}}.
\end{aligned}
\tag{9}
$$

Finding the Mapping X

- Now we have known for each element of $\phi(Z^{\text{in},i})$ what the corresponding linear index in $Z^{\text{in},i}$ is.
- Next we discuss the implementation details

A MATLAB Implementation I

- First, we compute elements in (6) with $j = 1$ by MATLAB's '+' operator
- This operator has implicit expansion behavior to compute the outer sum of two arrays.
- From the second part of (6), we calculate the outer sum of

$$\begin{bmatrix} 0 \\ \vdots \\ h-1 \end{bmatrix} d^{\text{in}} + 1$$

and

$$\begin{bmatrix} 0 & \dots & h-1 \end{bmatrix} a^{\text{in}} d^{\text{in}}$$

A MATLAB Implementation II

- The result is the following matrix

$$\begin{bmatrix} 1 & a^{\text{in}}d^{\text{in}}+1 & \dots & (h-1)a^{\text{in}}d^{\text{in}}+1 \\ d^{\text{in}}+1 & (1+a^{\text{in}})d^{\text{in}}+1 & \dots & (1+(h-1)a^{\text{in}})d^{\text{in}}+1 \\ \vdots & \vdots & \dots & \vdots \\ (h-1)d^{\text{in}}+1 & ((h-1)+a^{\text{in}})d^{\text{in}}+1 & \dots & ((h-1)+(h-1)a^{\text{in}})d^{\text{in}}+1 \end{bmatrix} \tag{10}$$

- If columns are concatenated, we get (6) with $j = 1$
- To get (7) for all channels $j = 1, \ldots, d^{\text{in}}$, we have

$$\begin{bmatrix} \text{vec}((10)) + 0 \\ \vdots \\ \text{vec}((10)) + d^{\text{in}} - 1 \end{bmatrix}$$

A MATLAB Implementation III

- This can be computed by the following outer sum:

$$\text{vec}((10)) + \begin{bmatrix} 0 & 1 & \ldots & d^{\text{in}} - 1 \end{bmatrix} \tag{11}$$

  Note that in (11) we have

$$0, \ldots, d^{\text{in}} - 1$$

  instead of

$$1, \ldots, d^{\text{in}}$$

  because in (10) we have already done "+1" for $j = 1$

A MATLAB Implementation IV

- Now we have linear indices of the first column of $\phi(Z^{\text{in},i})$
- Next, we obtain other columns in $\phi(Z^{\text{in},i})$
- In the linear indices in (9), the second term corresponds to indices of the first column, while the first term is the following column offset

$$(a + ba^{\text{in}})sd^{\text{in}}, \ \forall a = 0, 1, \ldots, a^{\text{out}} - 1,$$
$$b = 0, 1, \ldots, b^{\text{out}} - 1.$$

A MATLAB Implementation V

- This column offset is the outer sum of the following two arrays.

$$\begin{bmatrix} 0 \\ \vdots \\ a^{\text{out}} - 1 \end{bmatrix} \times s d^{\text{in}} \quad \text{and} \quad \begin{bmatrix} 0 & \ldots & b^{\text{out}} - 1 \end{bmatrix} \times a^{\text{in}} s d^{\text{in}} \tag{12}$$

- Finally, we compute the outer sum of
  - the column offset and
  - the linear indices in the first column of $\phi(Z^{\text{in},i})$

A MATLAB Implementation VI

in the following operation

$$\text{vec}((12))^T + \text{vec}((11)), \tag{13}$$

where

$$\text{vec}((12)) \in R^{a^{\text{out}} b^{\text{out}} \times 1} \text{ and } \text{vec}((11)) \in R^{hhd^{\text{in}} \times 1}$$

- In the end we store

$$\text{vec}((13)) \in R^{hhd^{\text{in}} a^{\text{out}} b^{\text{out}} \times 1}$$

It is a vector collecting

A MATLAB Implementation VII

column index of the non-zero in each row of P_ϕ^m

- Note that each row in the $0/1$ matrix P_ϕ^m contains exactly only one non-zero element.
- See the example in (5)
- The obtained linear indices are independent of the values of $Z^{\text{in},i}$.
- Thus the above procedure only needs to be run once in the beginning.

A Simple MATLAB Code I

```
function idx = find_index_phiZ(a,b,d,h,s)

first_channel_idx = ([0:h-1]'*d+1) +
                    [0:h-1]*a*d;
first_col_idx = first_channel_idx(:) + [0:d-1];
a_out = floor((a - h)/s) + 1;
b_out = floor((b - h)/s) + 1;
column_offset = ([0:a_out-1]' +
                [0:b_out-1]*a)*s*d;
idx = column_offset(:)' + first_col_idx(:);
idx = idx(:);
```

Discussion

- The code is simple and short
- We assume that Matlab operations used here are efficient and so is our resulting code
- But is that really the case?
- We will do experiments to check this
- Some works have tried to do similar things (e.g., `https://github.com/wiseodd/hipsternet`), though we don't see complete documents and evaluation