این فایل شامل توضیح کامل هر یک از تکنیک (الگوریتم) های مورد مطالعه است.

موارد مورد بررسی برای هر الگوریتم به شرح زیر خواهد بود.

**نام:** نام

**شرح مختصر:** شرح مختصر

هدف بهینه سازی: "هدف الگوریتم بهینه سازی چه چیزی است"

**نوع عملکرد:** "توضیح کامل چگونگی کارکرد الگوریتم ترجیحا با مثال"

**وابستگی:** "استفاده از این الگوریتم وا بسته به کدام الگوریتم ها است"

**تضاد:** "در صورت استفاده از این الگوریتم کدام الگوریتم ها را نمیتوان استفاده کرد"

**پیشنهاد:** "چه الگوریتم هایی در کنار این الگوریتم باعث بوجود آمدن نتایج بهتری خواهند شد"

**خنثی:** "در صورتی که از این الگوریتم استفاده کنیم استفاده از کدام الگوریتم ها بی معنی خواهد بود."

**نوع داده های ورود:** "این الگوریتم روی چه نوع داده هایی قابل اعمال است."

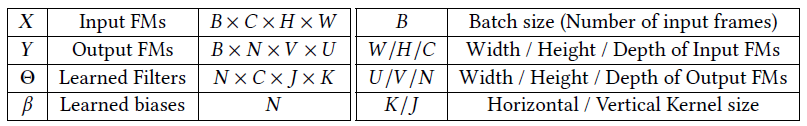
**بهینه ترین نوع داده ورودی:** "این الگوریتم با چه نوع داده ای در ورودی های خود عملکرد بهتری خواهد داشت."

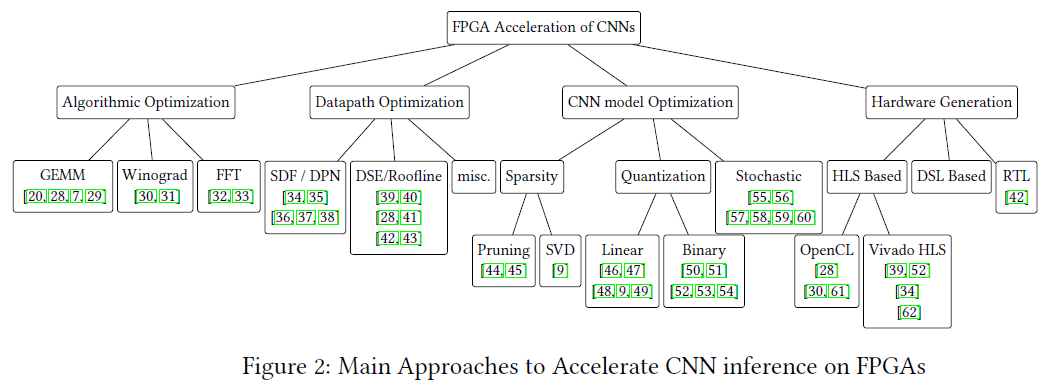
**نوع داده های خروجی:** "انواع داده هایی که میتواند در خروجی ایجاد کند."

**شبکه های استفاده شده:** "این الگوریتم در کدام یک از مدلها بکار گرفته شده است."

**بهینه ترین شبکه های استفاده شده:** "این الگوریتم در کدام یک از مدل های شبکه عصبی عملکرد بهتری داشته است."

**شبکه‌های پیشنهادی:** "پیشنهاد می‌شود این الگوریتم در کدام شبکه ها استفاده شود."





**بهینه سازی های سطح الگوریتم برای پیاده سازی شبکه های CNN روی FPGA**

1: computational transforms

به منظور شتابدهی اجرای لایه conv و FC از "computational transforms" روی FMs ها و Kernel ها استفاده میشود. این کار برای کاهش عملیات های ریاضی در هنگام استنتاج انجام میشود.

این " computational transforms" ها عمد تا روی CPU , GPU مورد استفاده قرار میگیرند و در قالب کتابخانه هایی مانند openblas برای cpu و cublas برای gpu موجود هستند. حالا از همین تبدیل ها برای map کردن CNN ها در FPGA استفاده میشود.

1.1: GEMM Transformation

در cpu,gpu یکی از روش های خیلی مرسوم استفاده از General Matrix Multiplications (GEMMs) است. OpenCL این رویکرد رو برای پیاده سازی های FPGA ارائه میکنه.{63,64}

