Institución Universitaria
Reacreditada en Alta Calidad

Neural Architecture Search

INTEGRACIÓN DE ML EN EMBEBIDOS Y EDGE COMPUTING

Somos Innovación Tecnológica con Sentido Humano

Alcaldía de Medellín

Contenido

1. Neural Architecture Search (NAS)
2. Métodos de Búsqueda
3. Búsqueda de arquitectura
4. Hardware-aware NAS
5. Otras técnicas

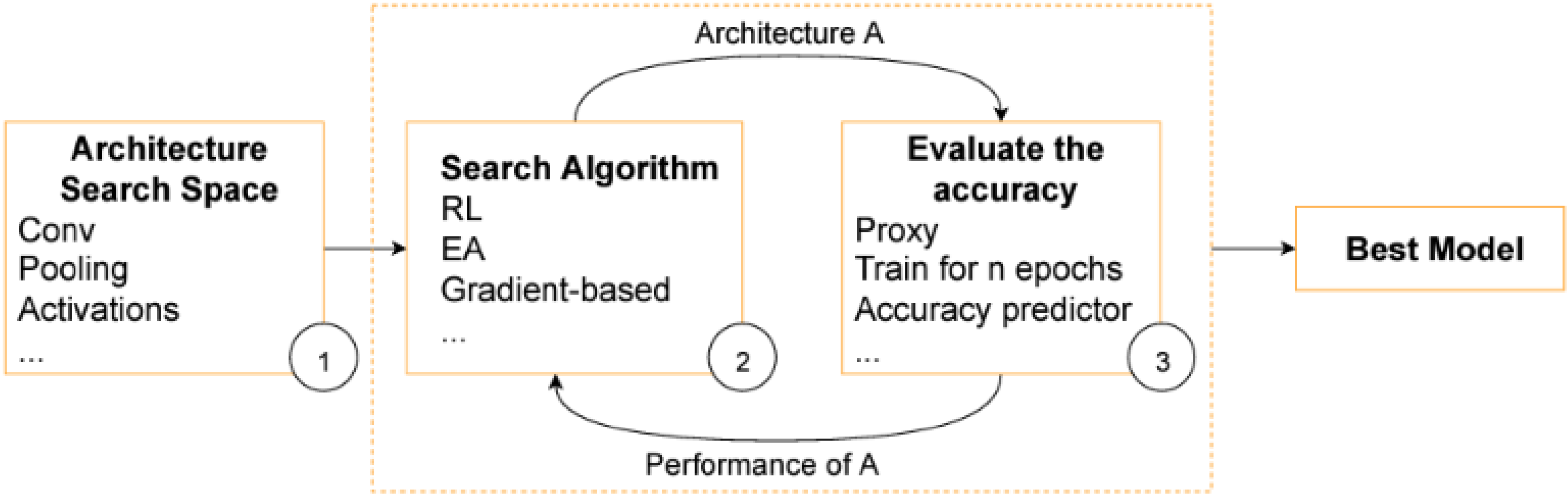
NAS – Neural Architecture Search

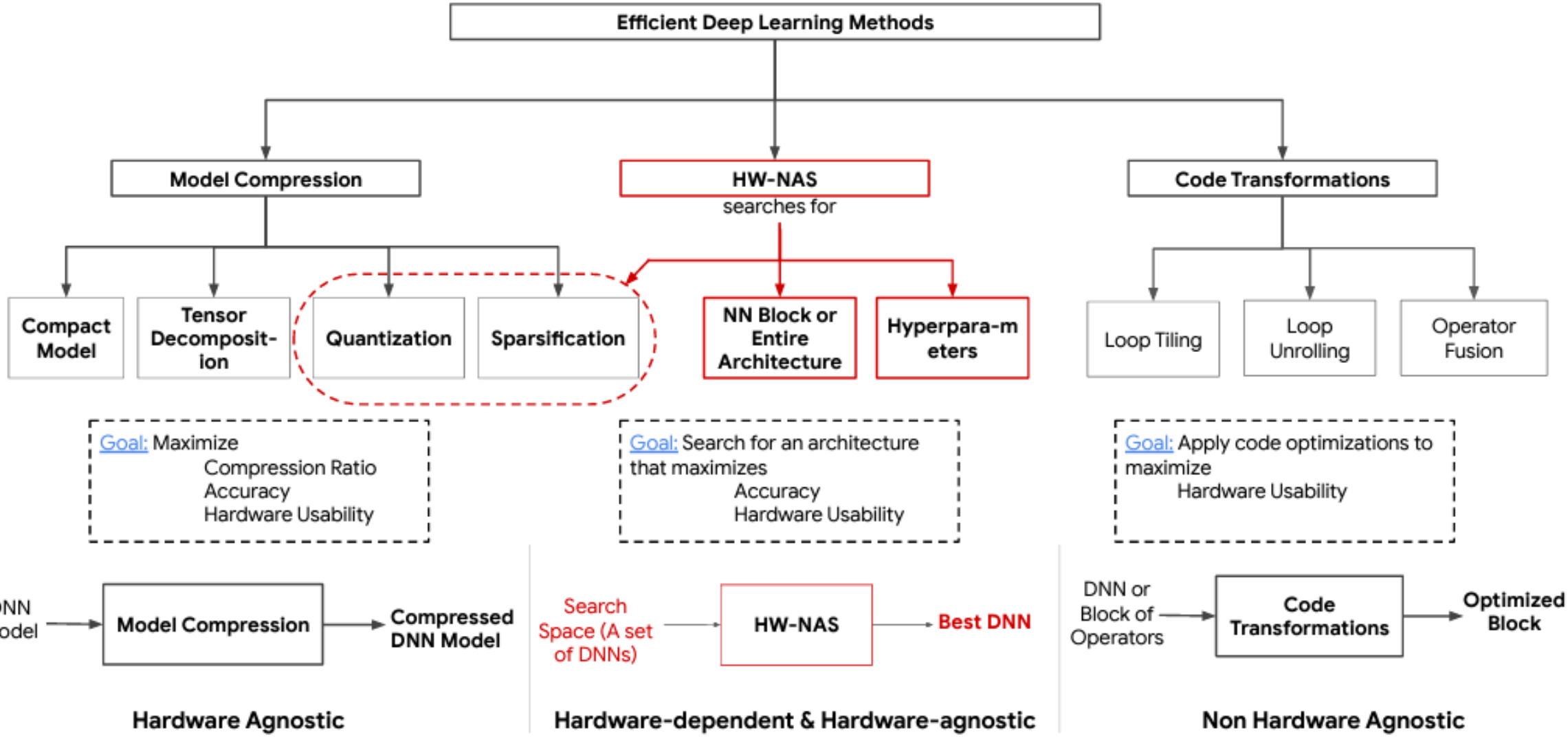
Un método NAS consiste de:

1. Espacio de búsqueda (Search Space): estructura y conjunto de operaciones primitivas construidas manualmente. Puede ser aplicado a celdas o a capas. Se definen los tipos de capas o redes que pueden se diseñadas y optimizadas.
2. Estrategia de búsqueda (Search Strategy): algoritmo de búsqueda a través del cual se busca una red en el espacio de búsqueda.
3. Fase de evaluación: donde la red encontrada es evaluada para su de desempeño.

Por lo general las técnicas NAS están enfocadas en encontrar la red neuronal que entregue mejor porcentaje de acierto o desempeño. Y estas soluciones llevan a redes neuronales gigantes. Se deben buscar son técnicas NAS para restricciones en hardware.

NAS



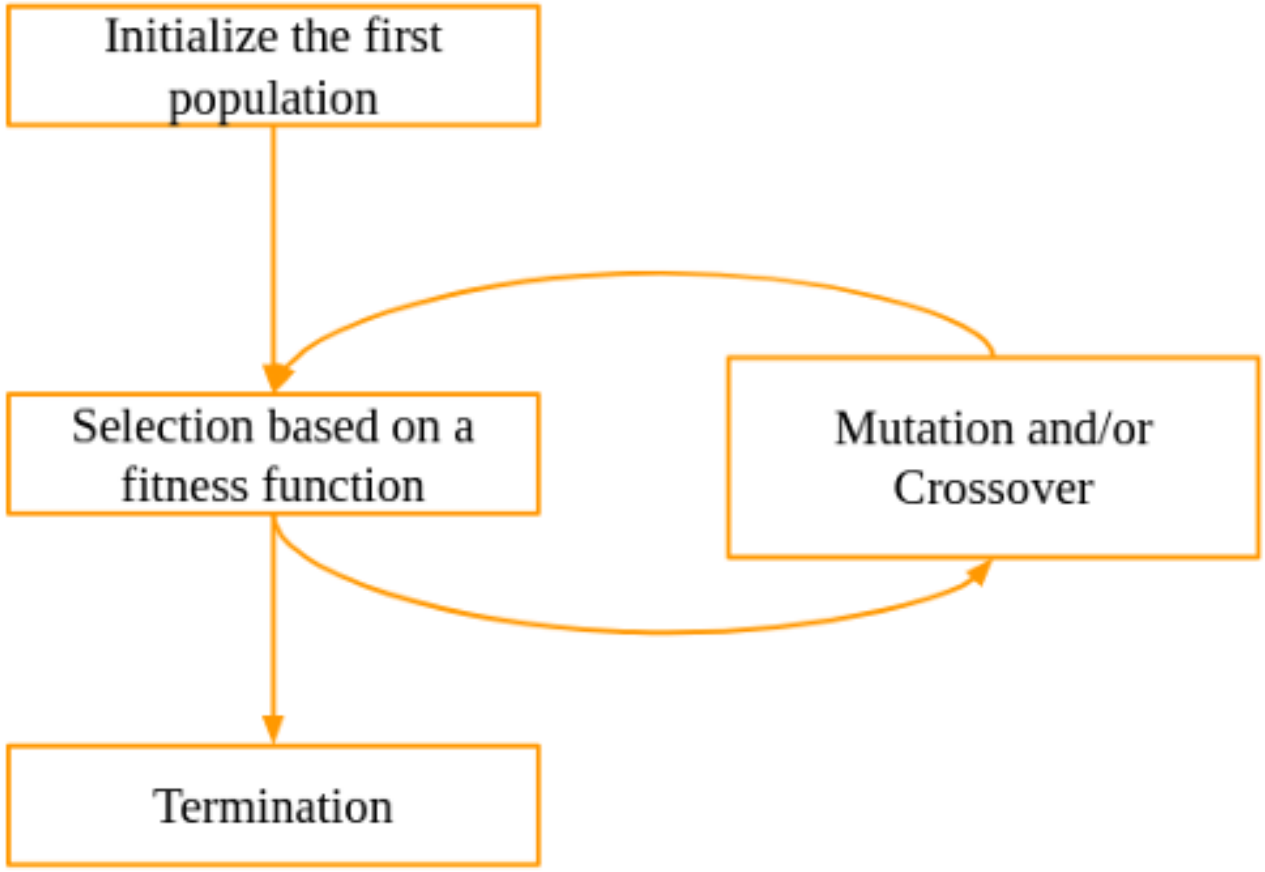


Contenido

1. Neural Architecture Search (NAS)
2. Métodos de Busqueda
3. Búsqueda de arquitectura
4. Hardware-aware NAS
5. Otras técnicas

Métodos de búsqueda - RL

Métodos de búsqueda - Optimización Heurística



Contenido

1. Neural Architecture Search (NAS)
2. Métodos de Búsqueda
3. Búsqueda de arquitectura
4. Hardware-aware NAS
5. Otras técnicas

Architectural Search Space

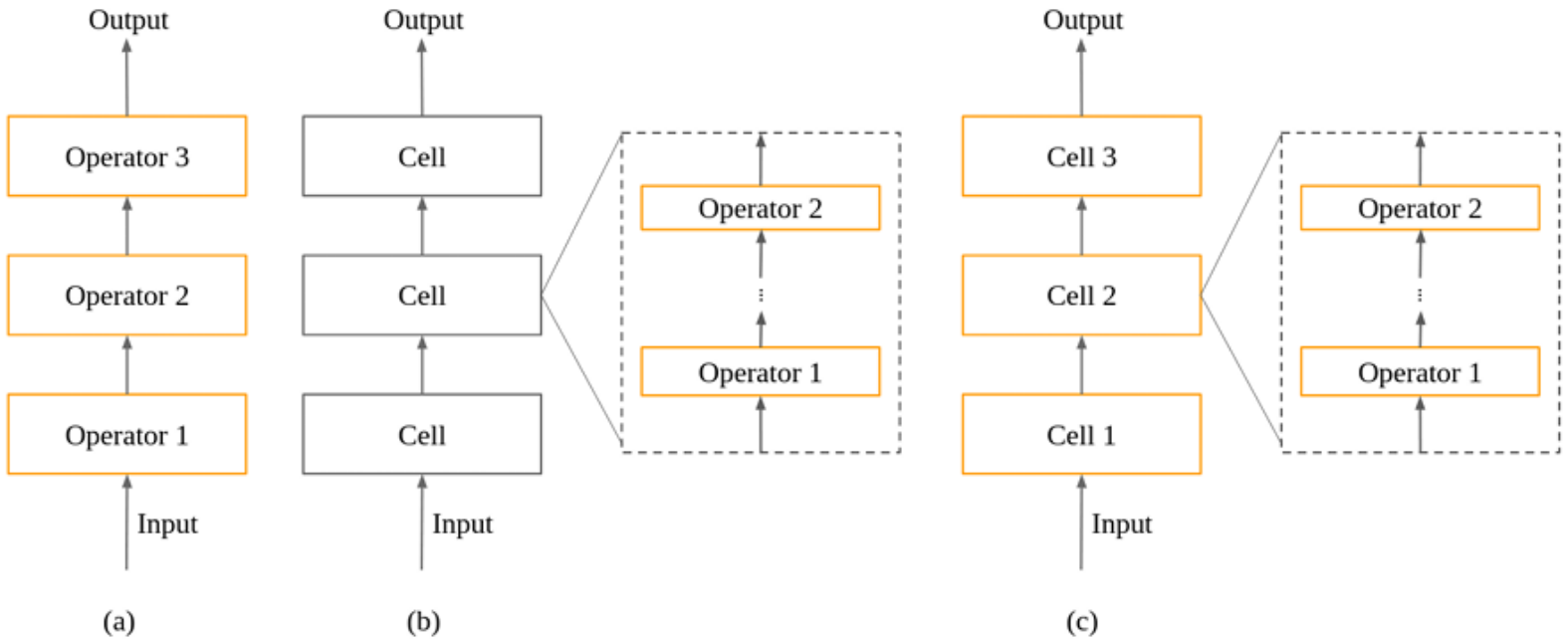


Fig. 10. Architecture search spaces types. (a) Global search space, (b) Cell-based search space, and (c) Hierarchical search space. In orange the operators considered during the search.

Contenido

1. Neural Architecture Search (NAS)
2. Métodos de Búsqueda
3. Búsqueda de arquitectura
4. Hardware-aware NAS
5. Otras técnicas

NAS – Constraint Hardware

Un método NAS para restricción de hardware es:

El proceso de optimización del diseño de la arquitectura red neuronal se debe enfocar en: espacio de memoria disponible, numero de FLOPs, numero de MACs (Multiply-ACumulate operations) o latencia de la inferencia.

Hardware-aware NAS

Problemas:

1. La cantidad de tipos variados de datos y tareas que requieren diferentes diseños y optimizaciones de arquitecturas neuronales.
2. La gran cantidad de plataformas de hardware que hacen difícil el diseño de una arquitectura globalmente eficiente.

Processor	Usecase	Compute	Memory	Power	Cost
Nvidia 1080Ti GPU [3]	Desktop	10 TFLOPs/Sec	11 GB	250 W	$700
Intel i9-9900K CPU [6, 5]	Desktop	500 GFLOPs/Sec	256 GB	95 W	$499
Google Pixel 1 (Arm CPU) [10]	Mobile	50 GOPs/Sec	4 GB	~ 5 W	–
Raspberry Pi (Arm CPU) [11]	Hobbyist	50 GOPs/Sec	1 GB	1.5 W	–
Micro:Bit (Arm MCU) [8]	IoT	16 MOPs/Sec	16 KB	~ 1 mW	$1.75
Arduino Uno (Microchip MCU) [1]	IoT	4 MOPs/Sec	2 KB	~ 1 mW	$1.14

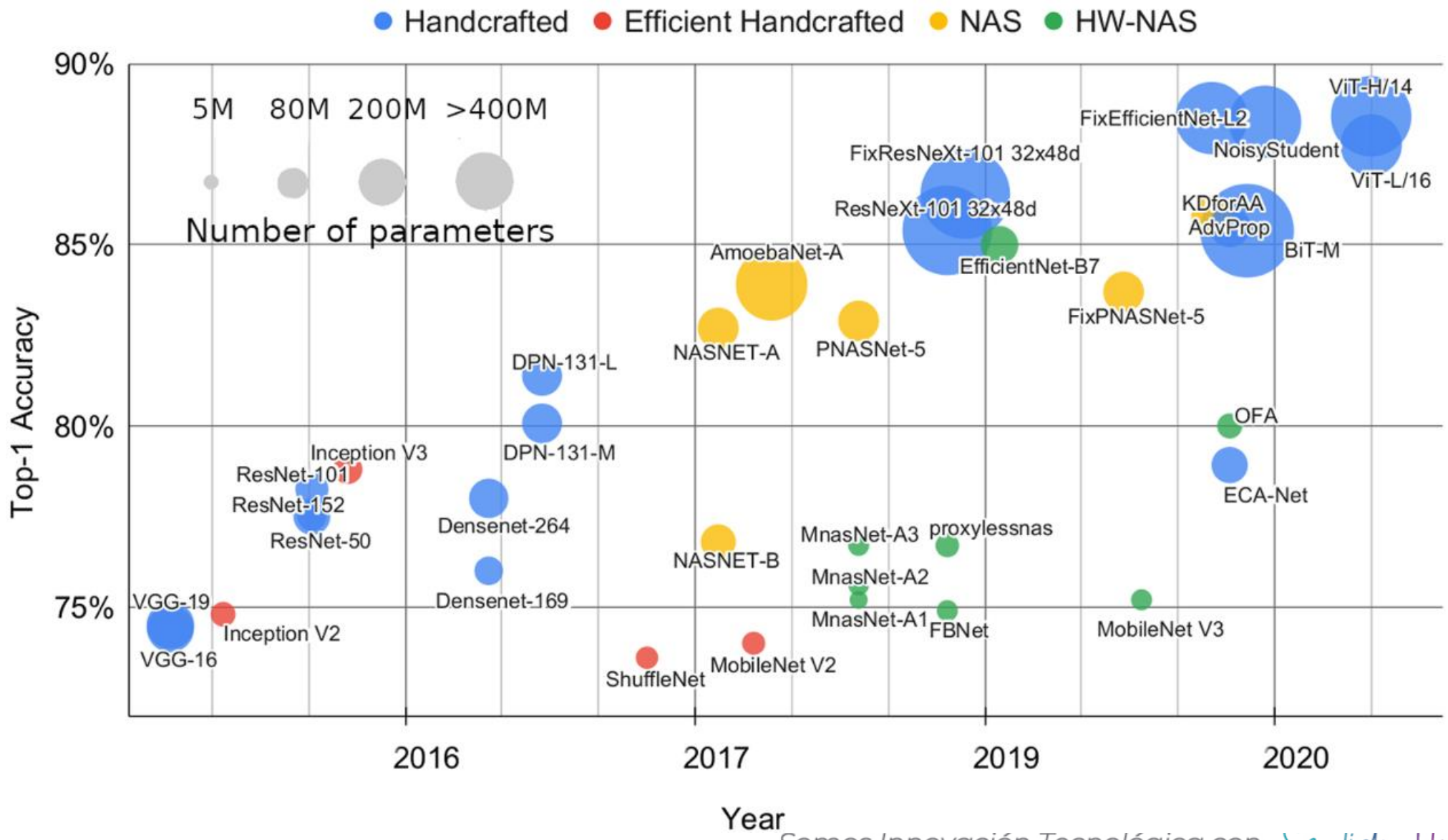
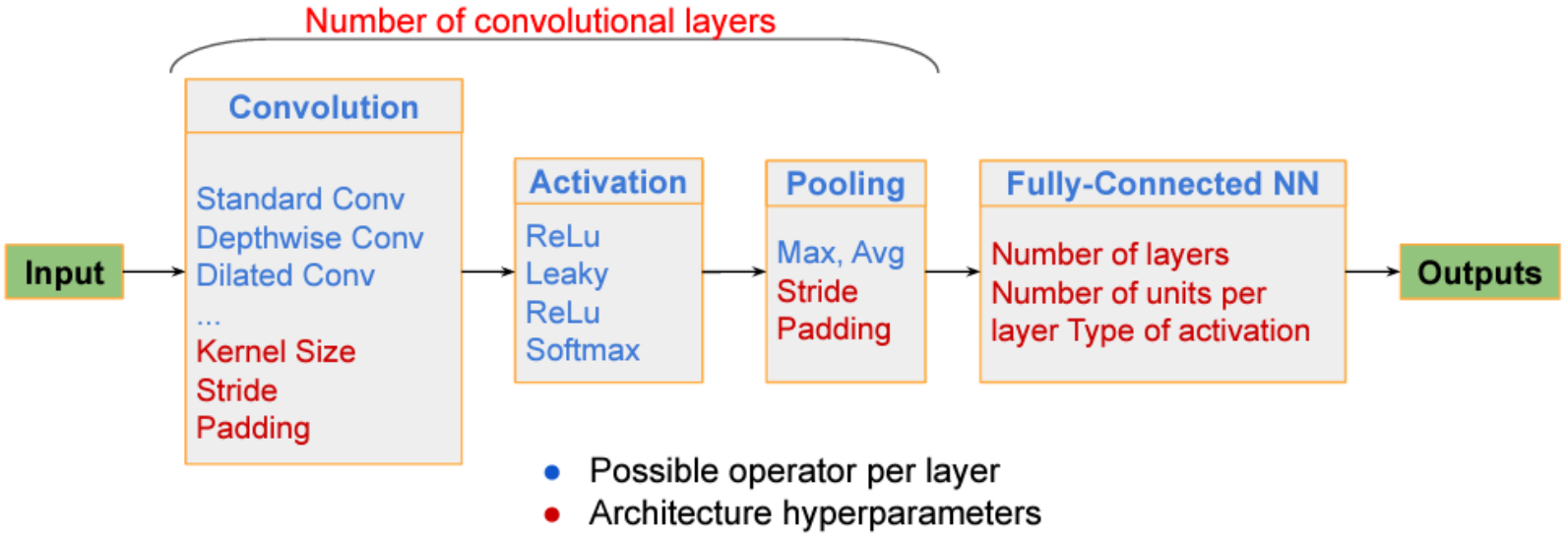


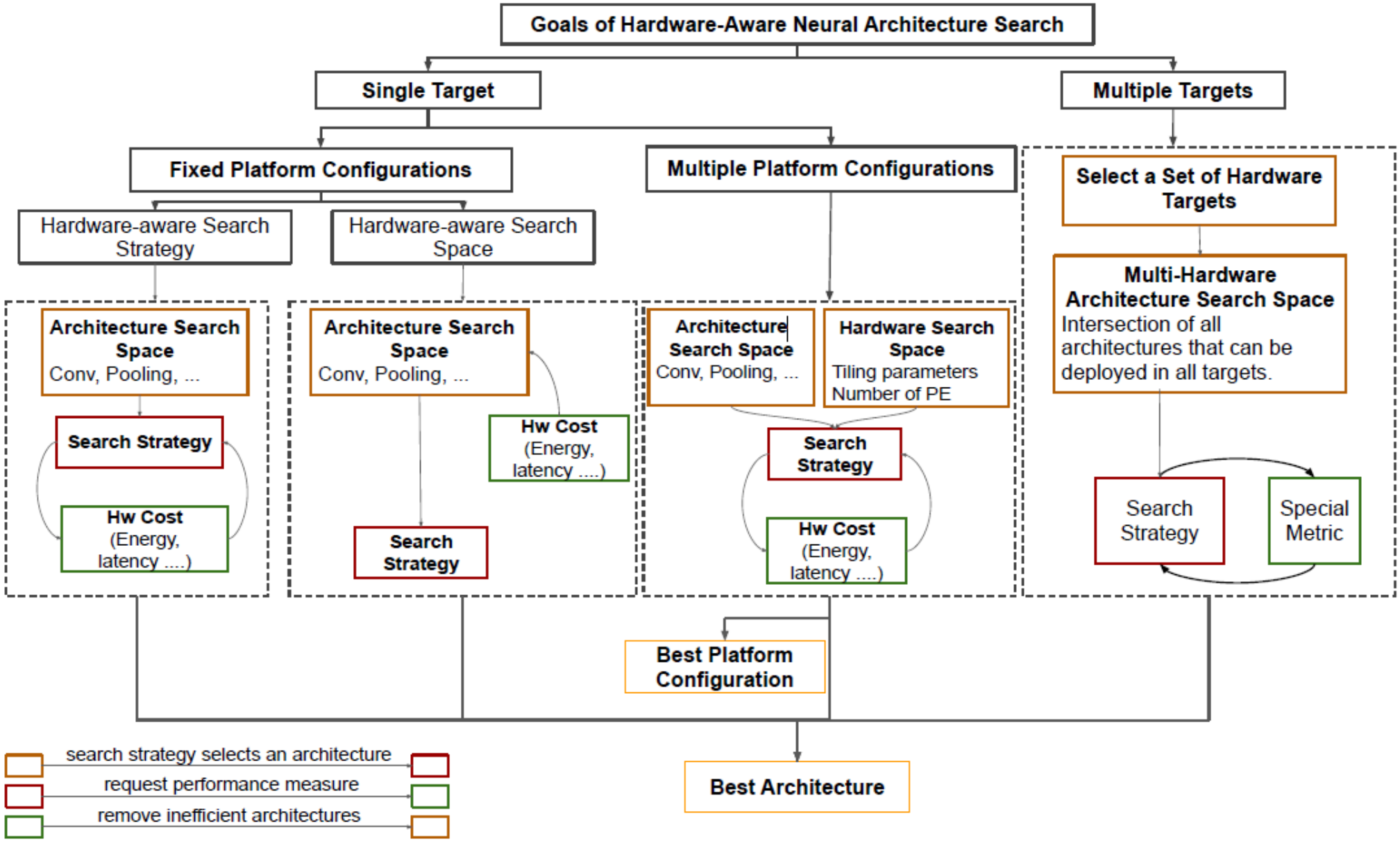
Tabla descriptiva de HW-NAS

Background	Efficient Deep Learning	Model Compression HW-NAS Code Transformations
	Algorithms	Reinforcement Learning Evolutionary Algorithm
Taxonomy of HW-NAS	Classification of HW-NAS based on their Goals	
Search Spaces	Architecture Search Space	
	Hardware Search Space	
HW-NAS Problem Formulation	Single-Objective Optimization	Two-Stage Search Constrained Optimization
	Multi-Objective Optimization	Scalarization NSGA-II
Search Strategies	Search Algorithm	Reinforcement Learning Evolutionary Algorithm Gradient-based Methods Bayesian Optimization & Random Search
	Non Differentiable Techniques	Over-parameterized Networks & their training
	Runtime Performance Optimization Strategies	Early Stopping Hot Start Proxy Datasets Accuracy Prediction Models
Hardware Cost Estimation Methods	Hardware Constraints Collection Techniques	Real-time measurements Lookup Table Analytical Estimation Prediction Models

Optimizar CNN



CNN generica. Para cada capa se selecciona un operador dentro de una lista predefinida (convolucion, maxpooling, batch_normalization…)



Contenido

1. Neural Architecture Search (NAS)
2. Métodos de Búsqueda
3. Búsqueda de arquitectura
4. Hardware-aware NAS
5. Otras técnicas

E-DNAS

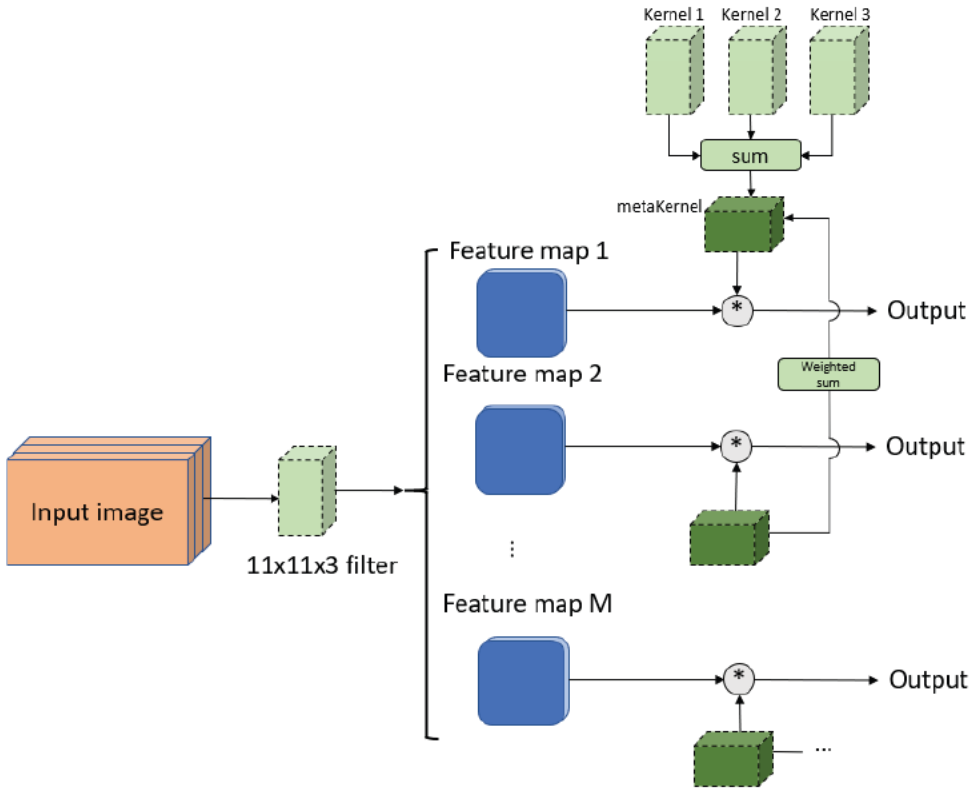


FIG. 1: **General overview of E-DNAS.** Our approach has two main building blocks: a depth-aware convolution with a high resolution 11×11 kernel followed by pairwise learning of meta-kernels with loopy flow of information on each iteration between training paths.

Algorithm 1: The search architecture methodology

Result: Find weights w_i and architecture probability parameters α to optimize the global loss function (13), given a defined search space with a combination of operations $\overline{o}^{(i,j)}$, defined in Eq. (9), a latency budget and an input dataset.

random initialization of α parameters

while *not converge* **do**

  Similar to [10], we generate the kernel candidates.
  Calculate Loss through Eq. (13).
  Calculate $\partial L/\partial w_a$ and $\partial L/\partial \alpha$.
  Update weights and architecture probability parameters α.
  Update Kernels using Eq. 3 and Eq. 5.

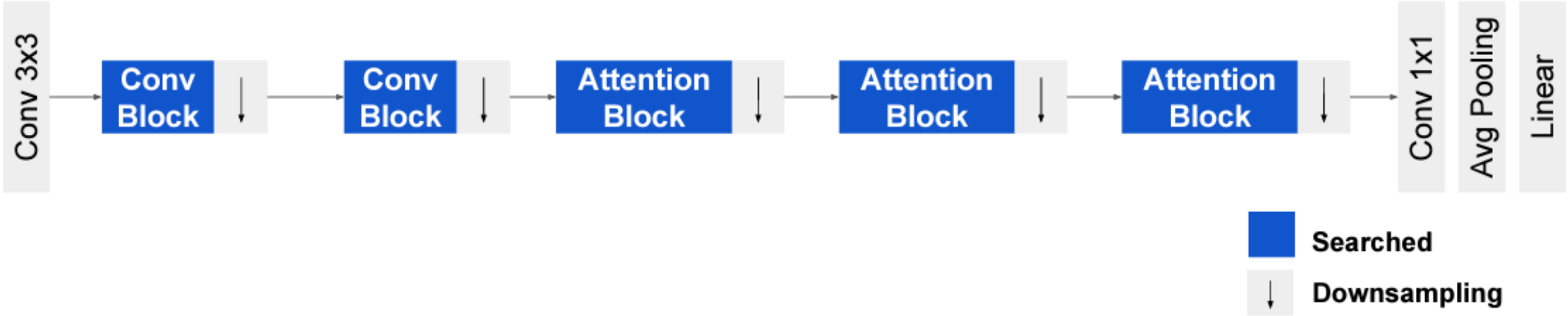
end

Extract more optimal architecture from learned α parameters.

squeezeNet

https://vitalab.github.io/article/2018/03/15/squeezeNet.html

HyT-NAS



Bibliografia

- H. Benmeziane et al. A Comprehensive Survey on Hardware-Aware Neural Architecture Search. 2021. https://arxiv.org/pdf/2101.09336.pdf
- https://github.com/alibaba/lightweight-neural-architecture-search

Institución Universitaria
Reacreditada en Alta Calidad

¡Gracias!

Somos Innovación Tecnológica con Sentido Humano

Alcaldía de Medellín