Created on	16/03/2021	Author	aiWare Engineering
Last modified	27/04/2021	Version	v3.0
Confidentiality:	Public	Document type	Technical Report

aiWare™ Benchmark Measurements

Technical Report

Created on	16/03/2021	Author	aiWare Engineering
Last modified	27/04/2021	Version	v3.0
Confidentiality:	Public	Document type	Technical Report

Contents

Created on	16/03/2021	Author	aiWare Engineering
Last modified	27/04/2021	Version	v3.0
Confidentiality:	Public	Document type	Technical Report

1 Introduction

AImotive's aiWare™ hardware IP offers silicon chip designers a highly scalable, fully synthesizable NN accelerator IP (referred to herein as an NPU – Neural Network Processing Unit) designed for very high efficiency real-time inference for automotive and other embedded safety-critical applications. It is optimized for large inputs from sensors such as cameras from VGA up to 8M pixels and beyond.

aiWare is independent of the type of the used neural network. Underpinned by its unique highly parallel scalable architecture, aiWare can process a wide range of CNNs with any level of depth and complexity, and with any input size.

aiWare is supported by a sophisticated SDK toolset that enables users to take a network trained in a floating-point domain and transform it to one using fixed point INT8 with little or no loss of accuracy. Tools include both coarse-grain and fine-grain transformation methodologies, complemented by dynamic hardware per-layer scaling that is used by the SDK compiler to optimize precision during every operation.

This document contains the results of detailed benchmark analysis performed by AImotive using the aiWare3P hardware IP core implemented using FPGA. Future revisions of this document will include comparison results where possible to other well-known NN accelerator platforms.

AImotive engineers have a long history of developing benchmarks. The predecessor company to AImotive, Kishonti Informatics, specialized in graphics, GPGPU and other system level benchmarks used by most major semiconductor companies, mobile phone OEMs and automotive Infotainment Tier1s. This experience has led to a rigorous benchmarking methodology, documented here, to ensure that all benchmark figures quoted here use a consistent test environment, are repeatable, and can be a trusted source of information for aiWare performance.

The aiWare benchmark documents are intended to demonstrate how aiWare performs when executing real-world automotive workloads, by focusing on defining the exact operating parameters and measurements using, wherever possible, real hardware. These benchmarks are based on measuring, overall efficiency - not utilization - of the NPU, since:

a) **Efficiency is the best way to assess any NPU's ability** to execute NNs: NPU Efficiency is the clearest measure of the ability of any NPU can execute a NN workload, based solely on its claimed maximum performance (in GMAC/s or TOPS).

b) NPU **Utilization is not a good measure of NPU performance**: the means by which it is calculated relies entirely on knowledge of the internal architecture of the NPU and means of measuring what it is doing. For example, if an NPU is using adders or multipliers to sometimes perform tasks such as data manipulation or address generation, this does not directly contribute to the execution of an NN but increases hardware utilization. In an extreme situation, if 50% of the time the ALUs within an NPU were being used for such tasks, the NPU vendor might claim 95% NPU utilization, but only achieve 45% NPU efficiency! By focusing on efficiency, these factors are removed, resulting in a much better indication of how well an NPUs claimed TOPS can be used for doing real work

This document is the first in a series of releases of results for these benchmarks, as we extend the benchmarking evaluation environment. In future releases, the measurement techniques will be further refined, and comparative analyses added with other commercially available automotive. Readers of this document are invited to submit to AImotive any suggestions for additional features for future releases via their regional salesperson.

1.1 Highlights of release v3.0

The following are the key things shown in this v3.0 release of this document:

- Initial measurements from first samples of the Nextchip Apache5 Imaging Edge Processor SoC, featuring a 1.6 TOPS, 1024 MACs aiWare3P implementation. For this initial report, measurements were performed using aiWare3P at 500MHz (i.e. 1.0 TOPS); later releases will update this data for 800MHz operation
- Measurements of a 4096 MAC aiWare3P implementation on FPGA
- Confirmation that aiWare can achieve better than 98% efficiency (not just hardware utilization – see 1.3) using a 2Mpixel and higher resolution input for:
  - Yolo v2
  - Resnet50
  - Resnet101
  - Resnet152
  - Vgg16
  - Vgg19
- Demonstration that for most benchmarks, aiWare can achieve 90% or better efficiency, and that efficiency usually improves with higher resolution inputs

1.2 Why are standard benchmarks not good enough?

Many engineers researching AI acceleration engines will use publicly available benchmarks as a starting point for comparisons. However for automotive applications, especially for vision-first systems relying on multiple high resolution cameras as inputs, these benchmarks can be misleading for several reasons:

i) The resolutions are often far too small – often 224 x 224. This is fine for certain applications, but they do not scale up well for cameras with orders of magnitude more data. When processing HD camera inputs, the performance challenge changes from being computation-dominated to data-dominated. Therefore, many NN accelerators claiming high TOPS performance will fail to maintain such performance once the significant additional data movement overheads of camera-based applications are taken into account.

ii) Most publicly available benchmark applications do not represent realistic automotive workloads. Many benchmarks come from identifying and classifying individual features such as one face, or distinguishing between a person, car, dog or sign. In automotive applications, we are trying to perceive as many different objects as possible. Most popular benchmarks do not reflect the demands of such NN workloads.

iii) Some publicly available benchmarks are not measuring sustained real-time throughput using a batch size of 1. Many NN accelerators achieve their highest efficiency by batching up at least 16 up to as many as 256 tasks and executing them simultaneously. Batching is not good for automotive, as it means latency is much higher as all tasks need to be completed before results are available. Automated driving applications, on the other hand, must minimize wherever possible the latency between when a sensor input is received, and the control output that results from those inputs – this is critical for both safety and performance, especially at high speeds. This is one significant example of where real-time NN inference is very different to mainstream general purpose ML, requiring NPUS to be designed and optimized for these requirements.

Furthermore, batching requires far more memory storage than batch=1 processing, as multiple frames of data must all be stored. In embedded systems, more memory means more cost and more power, so it must be kept as small as possible. This is especially true for automotive products intended for high volume production vehicles.

Because of these factors, and to achieve best performance for all benchmarks under realistic conditions, AImotive has modified public benchmarks in several ways:

- Performed benchmarks on a range of input sizes from 224 x 224 up to 8M (3840 x 2176) pixels
- Set batch size to 1 in all cases

1.3 NPU Efficiency is more important than NPU Utilization

The Efficiency for the execution of an NPU is the most appropriate method for measuring the ability of any NPU to execute any NN workload. It is defined as:

$$\frac{\text{Theoretical GMACs per Inference x IPS (Inferences Per Second)}}{\text{Capacity of the NPU in GMACs Per Second}}$$

The Utilization of an NPU only gives an arbitrary measure of how busy the NPU is when executing an NN, rather than how well it executes it. For aiWare, the Utilization and Efficiency are often very similar, since that is how aiWare was designed to operate. For other NPUs however, there can be significant differences, which demonstrate that Utilization is in practice a poor indication of actual NPU performance. Therefore, we strongly encourage everyone to use Efficiency as the primary method of evaluating the capabilities of any NPU.

Efficiency is a reliable and consistent measure for how good an NPU is at executing any given NN. Every NPU has its strengths and weaknesses, so we have chosen to use a wide range of different NN workloads to enable users to see the variation in performance for various NN topologies. We also include some of our own aiDrive NNs to demonstrate our efficiency when executing real automotive workloads rather than benchmarks that were often not developed to measure typical automotive applications.

We also use Efficiency to demonstrate how well an NPU manages data. In automotive applications, many of the most demanding workloads use high resolution sensor data as input, for example from the fusion of multiple different sensors, or from one or multiple high resolution camera sensors. We show all benchmarks over a range of input sizes, as this shows how the efficiency of an NPU scales with input size.

Created on	16/03/2021	Author	aiWare Engineering
Last modified	27/04/2021	Version	v3.0
Confidentiality:	Public	Document type	Technical Report

2 Benchmark Execution Environment

2.1 Benchmarking Methodology Overview

The aim of these benchmarks is to demonstrate aiWare3P's performance, and to help potential users the means to compare aiWare against other hardware architectures for executing automotive NN inference workloads, especially those applications where cameras, other high data rate sensors, or some form of heterogeneous large input data scenario such as early fusion are used as the primary inputs.

Whereas aiWare is designed primarily for low latency, real-time, embedded AI for camera-based automotive inference applications, most other architectures claiming to be suitable for automotive inference are in practice designed either for large batch size, non-real time training or inference in datacenters, or for small non-real time inference applications for mobile or IoT. Also, architectures such as GPUs have been designed to provide significant flexibility in computation across a wide range of algorithms, which has a significant impact on power, cost and performance compared to aiWare.

It is therefore the purpose of the benchmarking methodology adopted here to focus on relevant workloads for camera-centric NN inference for automotive applications, and the suitability of aiWare vs alternatives.

Measurements were taken based on the following assumptions:

- Input was available in accelerator's local memory prior to the start of execution
- Execution time was measured until the complete result was stored in the accelerator's local memory
- Power efficiency has not been measured

The following operation assumptions have been used:

- FPGA INT8 TOPS requirements = (INT8_multiplications + INT8_additions + INT8_comparisons)/10^9
- 2 INT8 TOPS = 1 TMAC/s$^{\text{(Note 1)}}$
- GPU TFLOP requirement = (FP32_multiplications + FP32_additions + FP32_comparisons)/10^9
- 2 FP32 TOPS = 1 FP32 TMAC/s
- MAC utilization = MAC_busy_time/total_inference_time
  = NN_MAC_requirement / (total_inference_time * MAC_capacity)

Note 1: only MACs used for convolution have been included in TOPS or TMAC/s for aiWare

2.1.1 INT8 Precision

All measurements on aiWare hardware use INT8, which is the native internal numerical precision of aiWare. However, thanks to hardware features such as per-layer scaling, 32-bit accumulation and other techniques embedded within the aiWare NPU, the highest possible precision has always been maintained. For NNs that were trained using FP32, the aiWare SDK was used to perform quantization, and checked to ensure that there was little or zero loss of precision when converting from FP32 to INT8.

2.1.2 Random Weights and Input Data

The runtime measurements were done on the above NNs with random weights and input data. The input data and weights were generated offline and loaded into the platform under test before runtime measurement commences. The output data of the network was not read back to keep measurements independent of host-dependent data movement overheads.

For some less-optimized networks, a significant number of weights for some layers have a value of zero, which can be advantageous for optimizing performance. In aiWare, the on-chip bandwidth compression feature ensures that any zero value data results in minimum overhead. However, it should be noted that for a well-optimized NN for production use, large amounts of zero value weights should be optimized out of the design, so in practice for a well-designed and well-optimized NN this will result in only modest benefits.

This is why the aiWare SDK is designed to be used by NN engineers, not DSP or embedded software engineers. By focusing on getting rapid performance analysis results using the aiWare Studio offline estimator, the NN engineer can quickly go back to their NN framework with the information needed to improve the design of the NN itself, because the information is shown in terms of layer execution, not just timing or other datapath-related analysis.

2.1.3 Input Resolution

NNs were run on a wide range of different input sizes ranging from 3x224x224 pixels to 3x3840x2176 pixels. For some networks, due to limitations of the prototype hardware platforms being used (often insufficient memory), not all resolutions could be measured for all NNs. This will be updated in future revisions of this document.

2.1.4 aiWare™ Tool Flow

All models used in these benchmarks use the standard Khronos NNEF format. The NNEF is exported directly from the framework used, usually Caffe, for the networks benchmarked here. The resulting NNEF representation is then compiled directly into a single binary that contains all network topologies, layer functionality, weights and aiWare scheduling commands.

For any network using ONNX, the ONNX network is first translated into NNEF using the standard Khronos ONNX to NNEF translator. Where ONNX has been used, it is identified in the benchmark; otherwise it should be assumed that standard NNEF exporters have been used from the framework itself.

All NNs used in this benchmark report can be downloaded from the AImotive website using the link in Appendix A.

2.2 Measurement Methodology

The primary method used for obtaining all detailed results in this report is detailed timing measurement of the workload executing on two hardware platforms: aiWare3P 1K integrated into the Nextchip Apache5 (1024 Convolution MACs) and aiWare3P 4K FPGA (4096 Convolution MACs) configurations.

2.2.1 Timing Measurement

All runtime measurements were done using the host PC's timer, with the value averaged over 50x network inferences. Over all measurements, the variance of the runtime values was less than 0.3%.

2.2.2 Correlation with aiWare™ Studio offline performance estimator

As part of our benchmarking process, every measurement made using physical hardware is compared with the equivalent performance estimation from the aiWare Studio Offline Performance Estimator. Based on this extensive analysis, our measurements indicate that our aiWare Studio offline performance estimator is accurate to within 2% for most workloads, and rarely more than 5%.

2.3 Hardware Environment

Measurements were carried out in the following hardware platforms:

- aiWare3P 1K Apache5: aiWare3P integrated onto the Nextchip Apache5 IEP SOC
  - MACs: 1,024; Clock speed: 500MHz; internal SRAM; 17.3 Mbits
  - Host CPU: Quad-core Arm A53

- aiWare3P 4K FPGA: aiWare3P hardware IP deployed on HTG-910 FPGA card
  - MACs: 4,096; Clock speed: 150MHz; internal SRAM: 17.3MBits
  - Host CPU: Intel i7-6700T 2.8 GHz with 64 GBytes main memory

2.3.1 Memory bandwidth

The results in this document assume that sufficient external memory bandwidth is available to the aiWare3P core to ensure the core is not stalled. Since aiWare3P compiler ensures that all external memory accesses are highly optimized to ensure maximum efficiency for every external memory transaction, we are confident that provided an appropriate external memory subsystem is used, these results should be achievable whether aiWare is integrated into an SoC using shared memory, or in a dedicated accelerator configuration with its own private external memory.

Since aiWare3P incorporates both external memory bandwidth compression and external data compression capabilities, we have included analysis with and without compression to demonstrate the performance of these capabilities.

To help users better understand the memory bandwidth requirements, the aiWare3P Studio Performance Analysis tools include detailed memory bandwidth estimation per layer. These advanced tools enable users to see the volume and timing of external memory transactions, and the total memory bandwidth required for any aiWare configuration for any workload.

2.4 Public NNs Used

The following publicly available neural networks were used in the measurements (please refer to Appendix A for details):

- GoogleNet
- vgg16
- vgg19
- Inception v4
- Inception_ResNet v2
- MobileNet v1
- ResNet50 v1
- ResNet101 v1
- ResNet152 v1
- Yolo v2

The descriptor files of all neural networks used in this benchmark can be downloaded. See Appendix A – References.

Note that for v3.0 of this document, only the base networks have been measured. However, in future releases we will include additional measurements showing the impact of adding NN heads such as Segmentation, and their potential impact on NPU performance.

2.5 AImotive aiDrive™ NNs used

We have used the following AImotive-developed NNs to evaluate the performance of aiWare in a more realistic automotive application environment. We believe that while public benchmarks provide some useful reference points when comparing NPUs, they rarely reflect real automotive workloads that have quite different requirements and characteristics.

The following AImotive NNs have been used to benchmark aiWare3P:

- Apollo_aiwfast: speed-optimized NN built for a smart rear-view camera application executing on a number of different SoC platforms. This version includes preliminary optimization for aiWare; further optimizations are expected in future releases of this workload
- Apollo_aiwprec: accuracy-optimized NN built for a smart rear-view camera application executing on a number of different SoC platforms. This version includes preliminary optimization for aiWare; further optimizations are expected in future releases of this workload

3 Notes for this v3.0 Release

This release v3.0 of the aiWare benchmark document should be read considering the following:

- This version focuses on measurements of two new hardware platforms: the Nextchip Apache5 and a large FPGA implementation
- All measurements have been performed on these platforms; no estimations or extrapolations have been used
- All measurements have been compared to the current version of the aiWare Studio offline performance estimator as part of the benchmarking process. Most estimated results were within 5% of measurements, with more than 2/3rds of all estimations accurate to within 2% of measured results
- The hardware platforms used both had limited external memory available, as well as other limitations. Later releases of this document will feature updated results once these limitations have been mitigated

3.1 Conclusions – Sections 4 & 5

The benchmark results in Sections 4 & 5 of this document are intended to demonstrate the following:

c) It is vital to assess the **<u>Efficiency</u>** of an NPU **<u>as a function of maximum input size</u>**. Since an increasing number of automotive automated driving solutions use cameras as either the primary sensor (e.g. Tesla, AImotive's aiDrive), or as a key input, it is essential that input size is a key evaluation parameter.

d) Since aiWare was designed to perform single or multi-sensor, large camera-first automotive inference, **<u>aiWare was designed from day 1 assuming that all data WILL NOT fit into on-chip memory</u>**, so the efficiency of aiWare often improves with larger data sizes, as any overheads relating to details such as boundaries, tiling etc. are minimized. This contrasts with many other NPUs that have been designed to work best using relatively small input sizes, whose performance assumes that **<u>most, if not all, of the DNN workload WILL fit into on-chip memory</u>**. Once this limit has been exceeded, their efficiency often decreases dramatically (for example with larger input sizes, or more complex DNNs with more channels

e) For aiWare, **<u>100% of all layer functions are executed by the aiWare NPU itself</u>**; the host CPU is never required during execution of any of the benchmarks shown here. The only functions the host CPU needs to perform are to initialize aiWare with all CNNs to be executed, supply input data, read output data, and respond to any error events

4 Benchmark Measurements – aiWare3P on Apache5 SoC

The Apache5 SoC from Nextchip is an advanced IEP (Imaging Edge Processor) for high performance, AI-enabled automotive edge applications. It is designed to full AEC-Q100 standards, operating over the full Grade 2 temperature range (-40C to +105C).

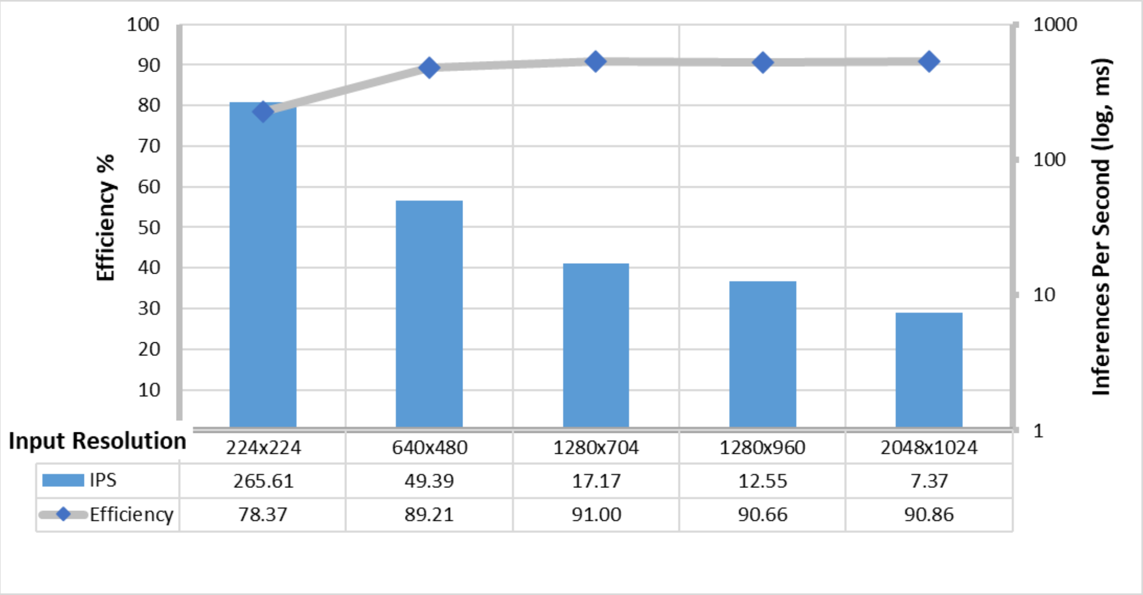
The Apache5 features an aiWare3P NPU IP core with the following key characteristics:

- Rated Performance: up to 1.6 TOPS
- 1,024 MACs
- Clock speed up to 800MHz
- 17.3 Mbits dedicated on-chip SRAM
- Full speed operation over full extended temperature range of -40C to +105C

For this release of this document, only the early Apache5 ES silicon is available, with limited memory. Tests have therefore been performed at 500MHz rather than 800MHz, and not all resolutions have been benchmarked due to lack of sufficient external DRAM on the board available to AImotive. Later revisions of this document will include results for 800MHz operation.

4.1 aiDrive™ Apollo_aiwprec

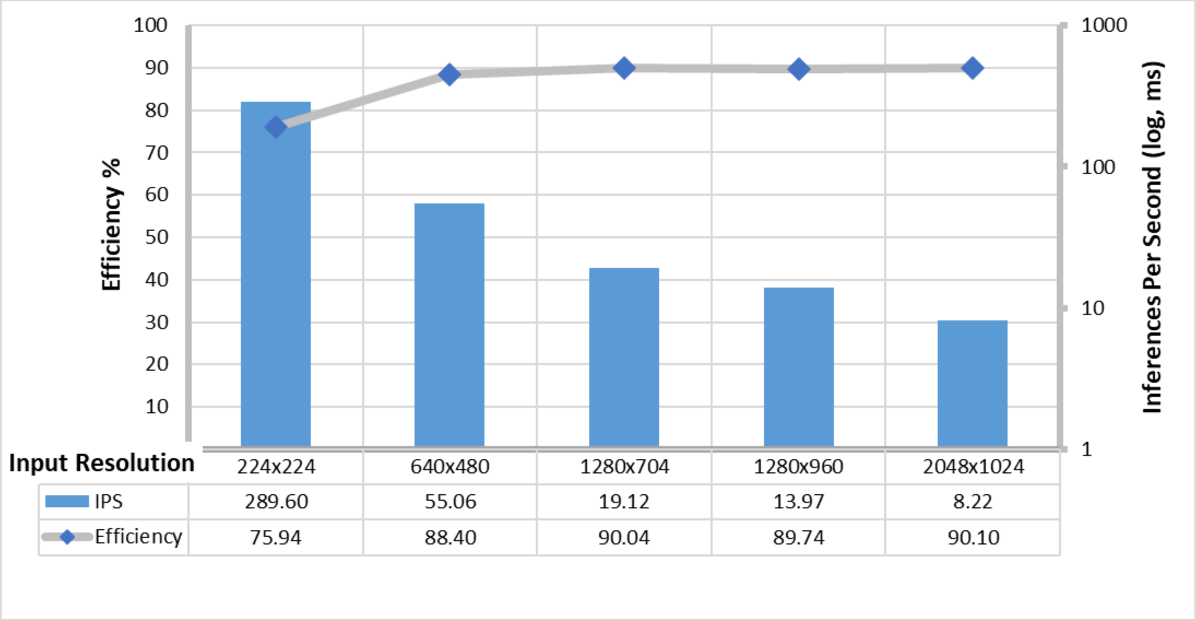
Apollo_aiwprec is an accuracy-optimized aiDrive™ automotive grade NN designed by AImotive for IRC (Intelligent Rear-View Camera) applications.



Input Resolution	224x224	640x480	1280x704	1280x960	2048x1024
■ IPS	265.61	49.39	17.17	12.55	7.37
◆ Efficiency	78.37	89.21	91.00	90.66	90.86

4.2 AIDRIVE™ APOLLO_AIWFAST

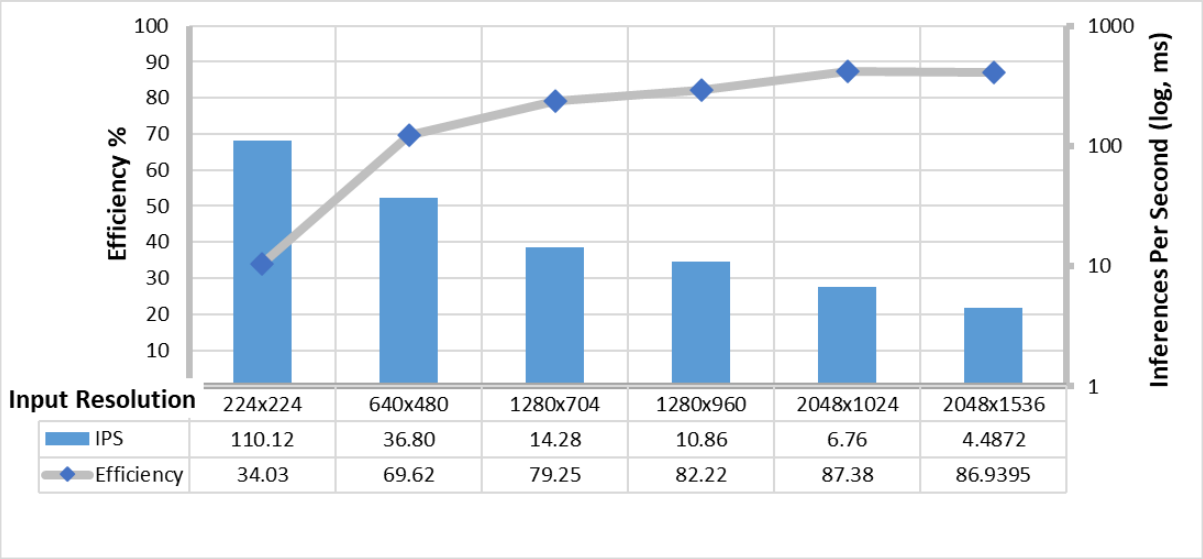
Apollo_aiwfast is a speed-optimized aiDrive™ automotive grade NN designed by AImotive for IRC (Intelligent Rear-View Camera) applications.



Input Resolution	224x224	640x480	1280x704	1280x960	2048x1024
▬ IPS	289.60	55.06	19.12	13.97	8.22
◆ Efficiency	75.94	88.40	90.04	89.74	90.10

4.3 GOOGLENET

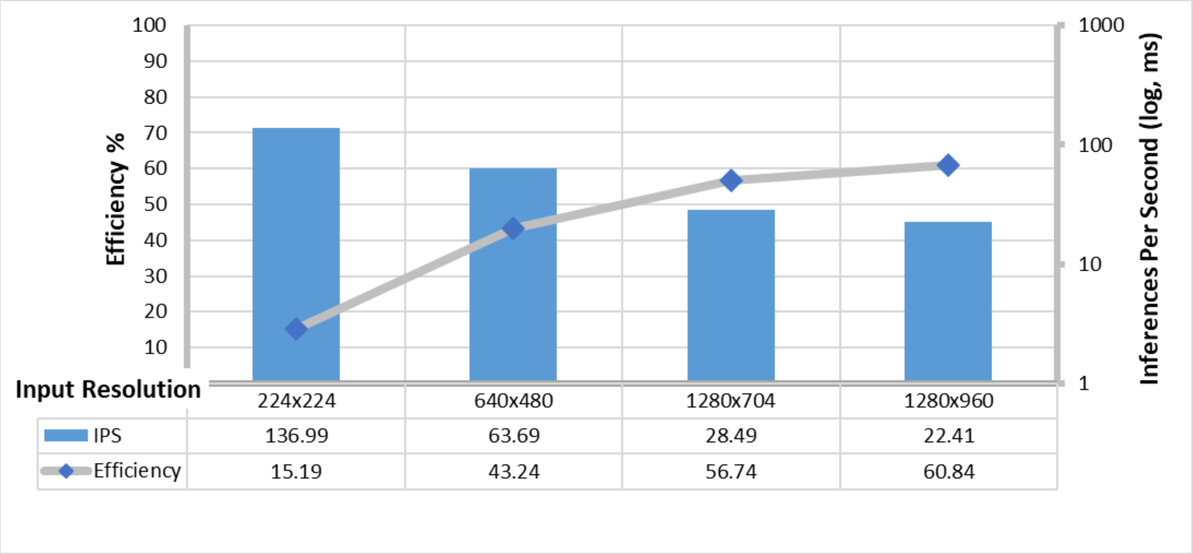
GoogleNet is a widely benchmarked public NN used in the ImageNet benchmark and is also used as a base network for other use cases. It was developed with the efficient Inception blocks. An inception block uses multiple filter sizes in parallel to efficiently process its input at different scales. We use its base network without the fully connected layers at the end, as ImageNet classification is not an applicable use case in automotive.



Input Resolution	224x224	640x480	1280x704	1280x960	2048x1024	2048x1536
▬ IPS	110.12	36.80	14.28	10.86	6.76	4.4872
◆ Efficiency	34.03	69.62	79.25	82.22	87.38	86.9395

4.4 MobileNet v1

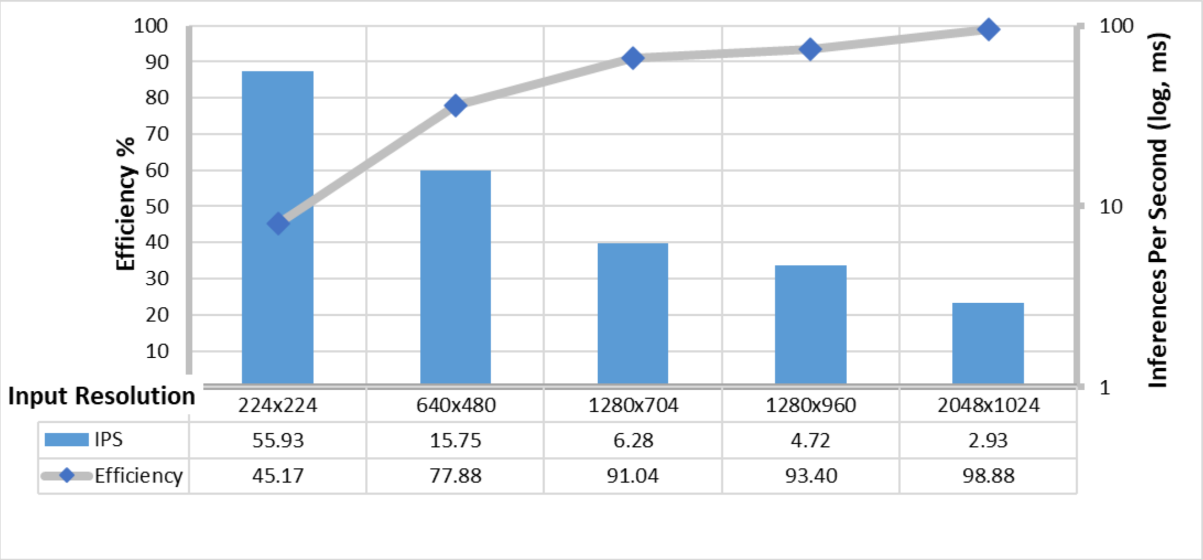
MobileNet is a widely benchmarked public NN used in the ImageNet benchmark and is also used as a base network for other use cases. It was developed with depth-wise separable convolutions, optimized mainly for computationally weak mobile processors. We use its base network without the fully connected layers at the end, as ImageNet classification is not a relevant use case in automotive.



Input Resolution	224x224	640x480	1280x704	1280x960
IPS	136.99	63.69	28.49	22.41
Efficiency	15.19	43.24	56.74	60.84

4.5 Yolo v2

Yolo v2 is a successful custom NN developed for bounding box object detection.



Input Resolution	224x224	640x480	1280x704	1280x960	2048x1024
IPS	55.93	15.75	6.28	4.72	2.93
Efficiency	45.17	77.88	91.04	93.40	98.88

5 Benchmark Measurements – aiWare3P on FPGA

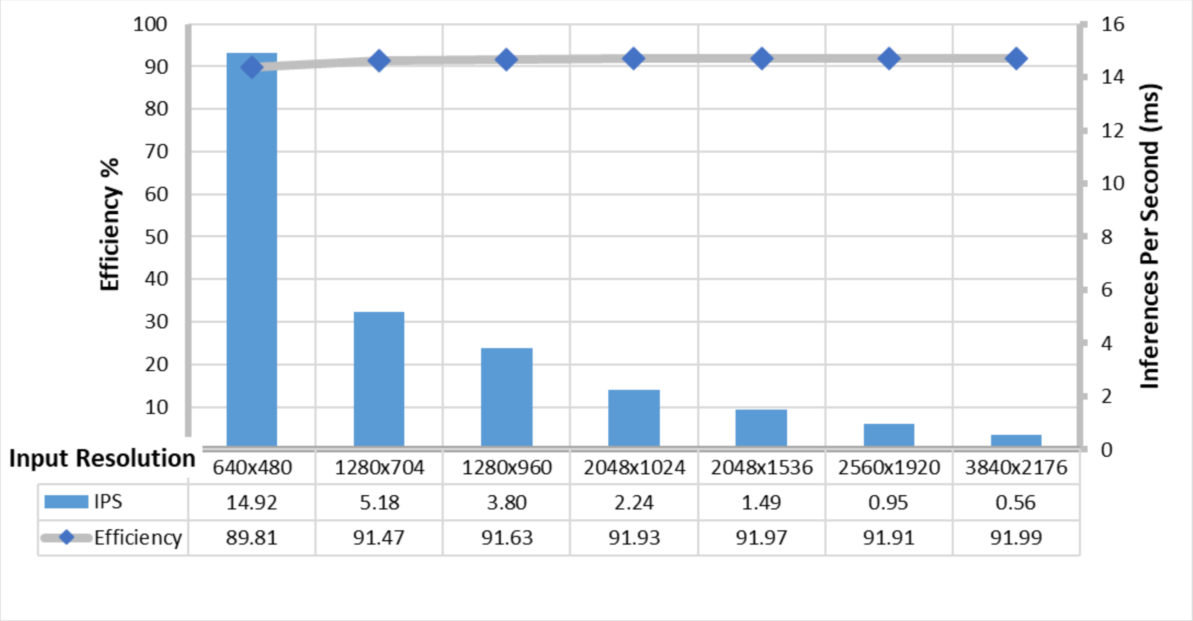
This section describes detailed measurements performed on FPGA hardware containing a production-quality implementation of the aiWare3P RTL. This FPGA implements an aiWare3P configuration with the following key characteristics:

- Rated Performance: 1.23 TOPS
- 4,096 MACs
- Clock speed of 150MHz
- 17.3 Mbits dedicated on-chip SRAM

For this release of this document, the clock speed an SRAM are limited. Future versions of this document will include larger configurations. However this demonstrates the scalability of aiWare to large numbers of MACs without compromising efficiency.

5.1 aiDrive™ Apollo_aiwprec

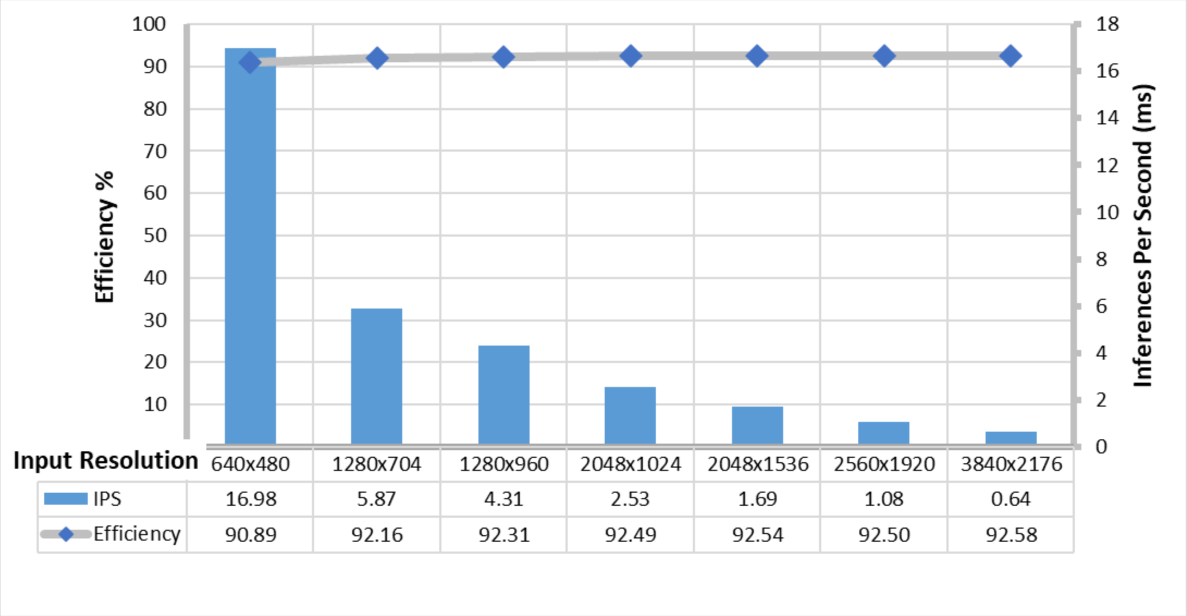
Apollo_aiwprec is an accuracy-optimized aiDrive™ automotive grade NN designed by AImotive for IRC (Intelligent Rear-View Camera) applications.



Input Resolution	640x480	1280x704	1280x960	2048x1024	2048x1536	2560x1920	3840x2176
IPS	14.92	5.18	3.80	2.24	1.49	0.95	0.56
Efficiency	89.81	91.47	91.63	91.93	91.97	91.91	91.99

5.2 aiDrive™ Apollo_aiwfast

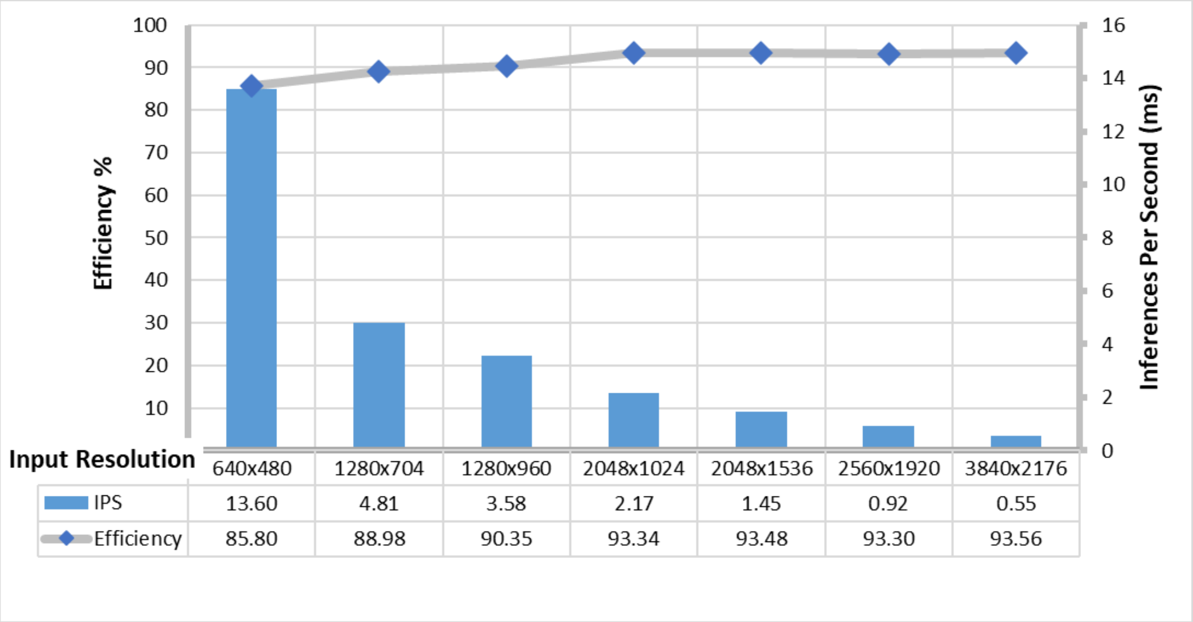
Apollo_aiwfast is a speed-optimized aiDrive™ automotive grade NN designed by AImotive for IRC (Intelligent Rear-View Camera) applications.



Input Resolution	640x480	1280x704	1280x960	2048x1024	2048x1536	2560x1920	3840x2176
IPS	16.98	5.87	4.31	2.53	1.69	1.08	0.64
Efficiency	90.89	92.16	92.31	92.49	92.54	92.50	92.58

5.3 GoogLeNet

GoogleNet is a widely benchmarked public NN used in the ImageNet benchmark, and is also used as a base network for other use cases. It was developed with the efficient Inception blocks. An inception block uses multiple filter sizes in parallel to efficiently process its input at different scales. We use its base network without the fully connected layers at the end, as ImageNet classification is not an applicable use case in automotive.

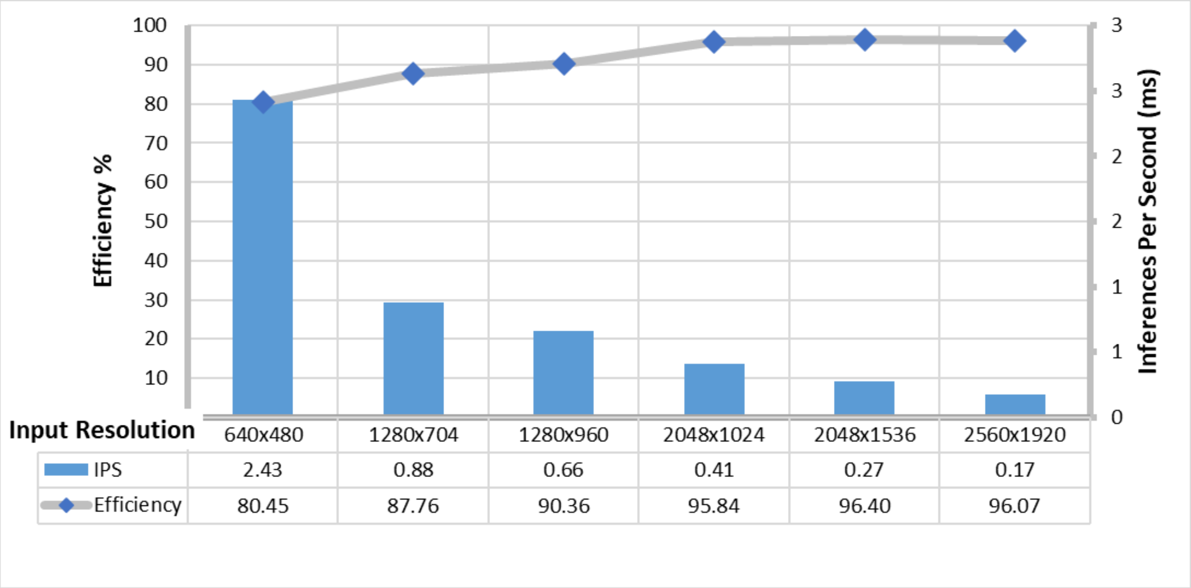


Input Resolution	640x480	1280x704	1280x960	2048x1024	2048x1536	2560x1920	3840x2176
IPS	13.60	4.81	3.58	2.17	1.45	0.92	0.55
Efficiency	85.80	88.98	90.35	93.34	93.48	93.30	93.56

Created on	16/03/2021	Author	aiWare Engineering
Last modified	27/04/2021	Version	v3.0
Confidentiality:	Public	Document type	Technical Report

5.4 Inception-ResNet v2

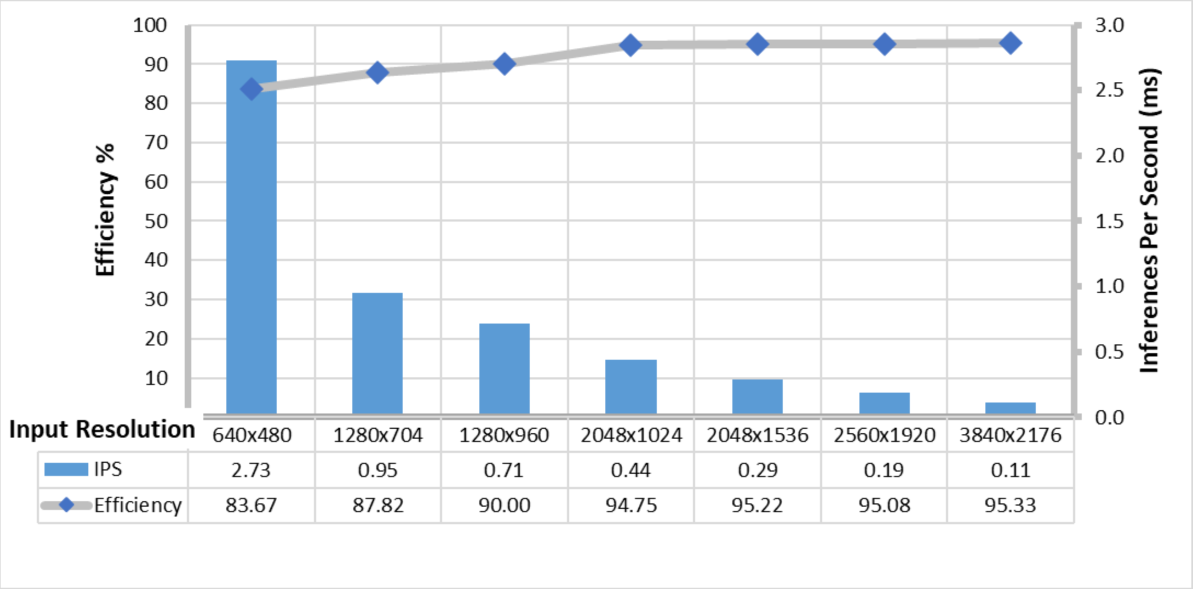
Inception-Resnet is a widely benchmarked public NN used in the ImageNet benchmark, and is also used as a base network for other use cases. It was developed with improved, more efficient Inception blocks and residual inception blocks. We use its base network without the fully connected layers at the end, as ImageNet classification is not a relevant use case in automotive.



Input Resolution	640x480	1280x704	1280x960	2048x1024	2048x1536	2560x1920
IPS	2.43	0.88	0.66	0.41	0.27	0.17
Efficiency	80.45	87.76	90.36	95.84	96.40	96.07

5.5 Inception v4

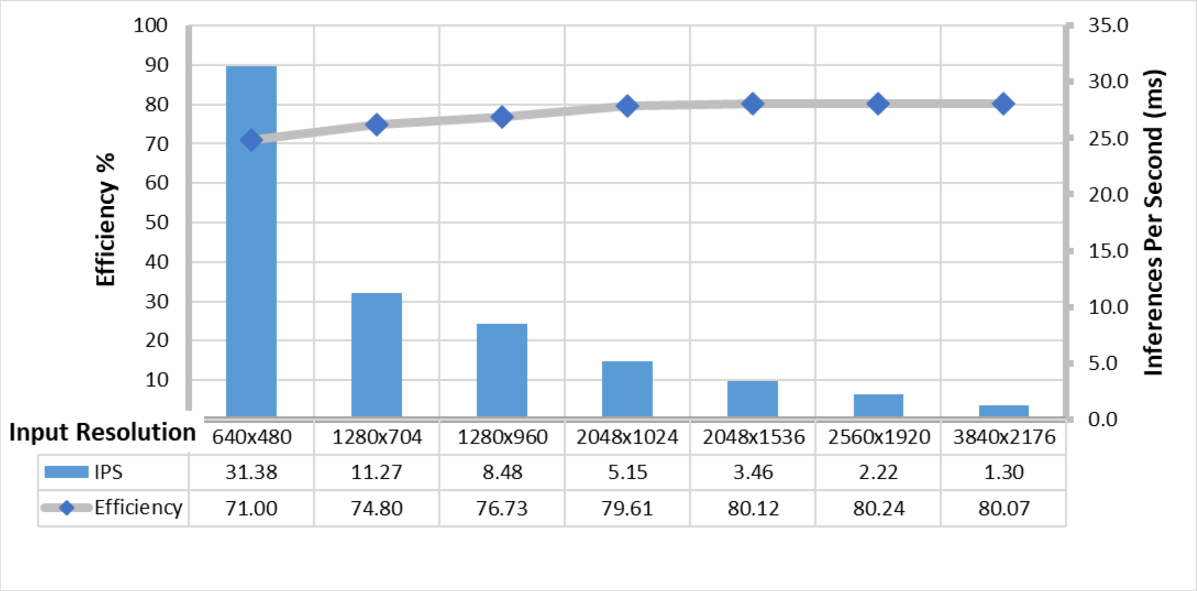
Inception_v4 is a widely benchmarked public NN used in the ImageNet benchmark, and as a base network for other use cases, too. It was developed with improved, more efficient Inception blocks, using asymmetric filter sizes. We use its base network without the fully connected layers at the end, as ImageNet classification is not a relevant use case in automotive.



Input Resolution	640x480	1280x704	1280x960	2048x1024	2048x1536	2560x1920	3840x2176
IPS	2.73	0.95	0.71	0.44	0.29	0.19	0.11
Efficiency	83.67	87.82	90.00	94.75	95.22	95.08	95.33

5.6 MobileNet

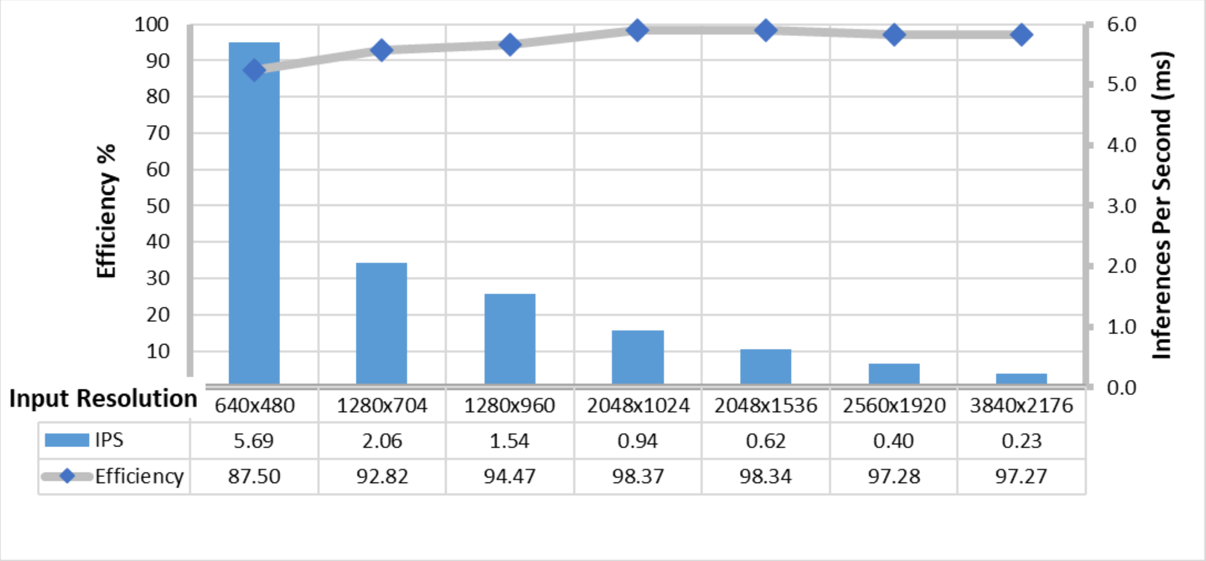
Mobilenet is a widely benchmarked public NN used in the ImageNet benchmark, and is also used as a base network for other use cases. It was developed with depthwise separable convolutions, optimized mainly for computationally weak mobile processors. We use its base network without the fully connected layers at the end, as ImageNet classification is not a relevant use case in automotive.



Input Resolution	640x480	1280x704	1280x960	2048x1024	2048x1536	2560x1920	3840x2176
IPS	31.38	11.27	8.48	5.15	3.46	2.22	1.30
Efficiency	71.00	74.80	76.73	79.61	80.12	80.24	80.07

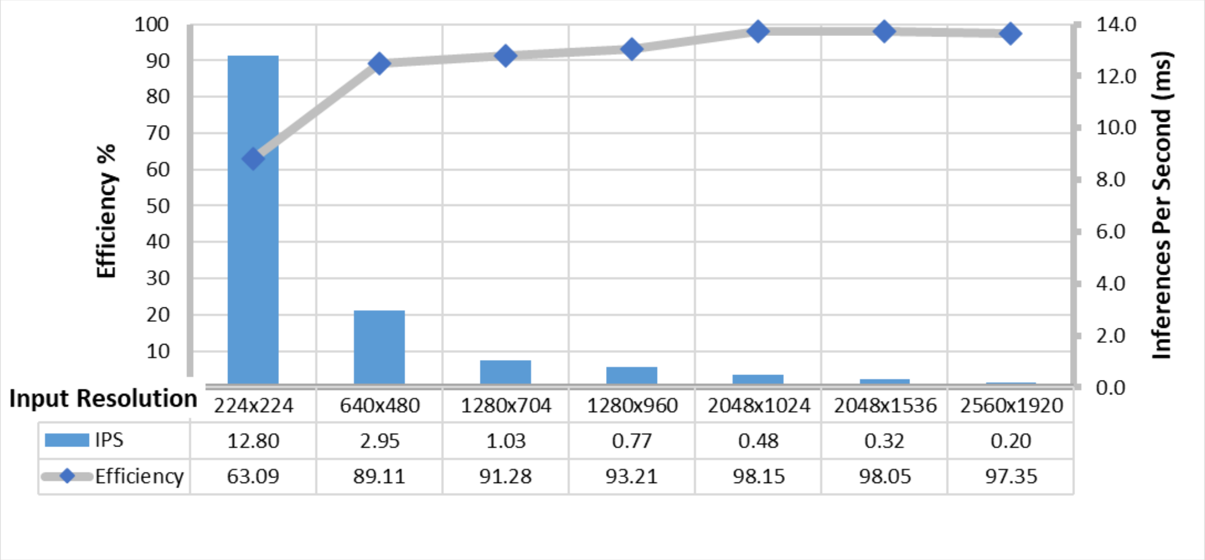
5.7 ResNet50 v1

Resnet50, Resnet101 and Resnet152 are widely benchmarked public NNs used in the ImageNet benchmark, and as a base network for other use cases. It was developed with residual connections, developed for demonstrating learning in very deep networks. We use its base network without the fully connected layers at the end, as ImageNet classification is not a relevant use case in automotive.



Input Resolution	640x480	1280x704	1280x960	2048x1024	2048x1536	2560x1920	3840x2176
IPS	5.69	2.06	1.54	0.94	0.62	0.40	0.23
Efficiency	87.50	92.82	94.47	98.37	98.34	97.28	97.27

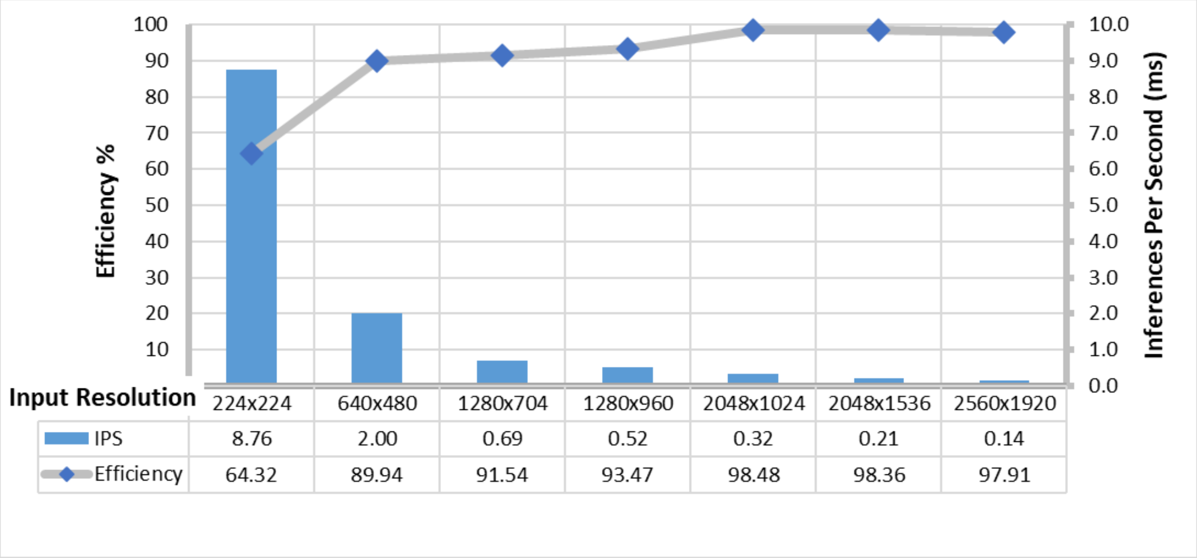
5.8 ResNet101 v1



Input Resolution	224x224	640x480	1280x704	1280x960	2048x1024	2048x1536	2560x1920
IPS	12.80	2.95	1.03	0.77	0.48	0.32	0.20
Efficiency	63.09	89.11	91.28	93.21	98.15	98.05	97.35

5.9 ResNet152 v1

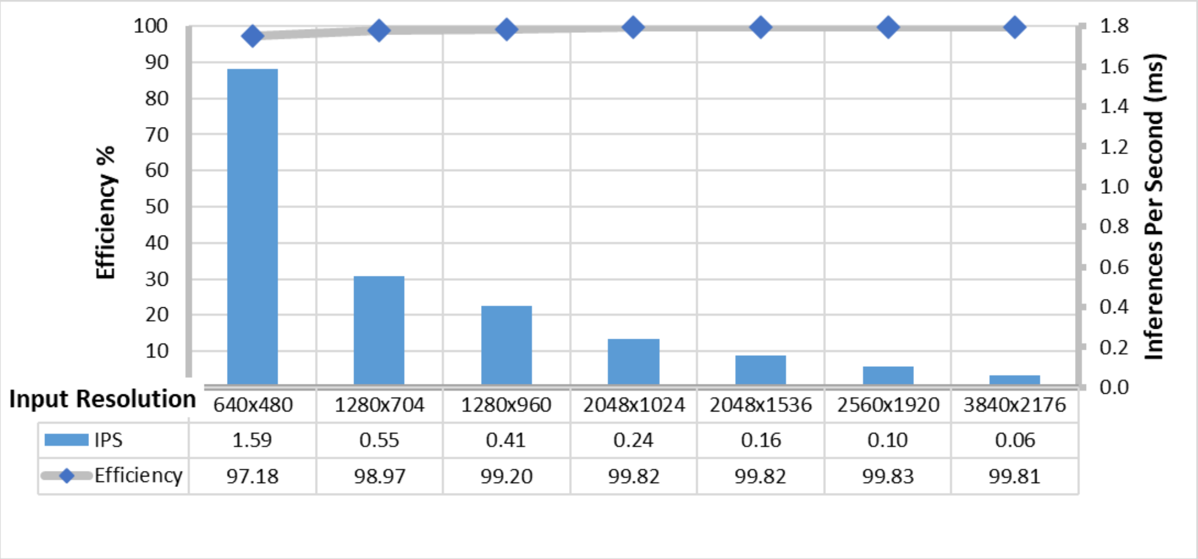
This is a less well-known public domain benchmark.



Input Resolution	224x224	640x480	1280x704	1280x960	2048x1024	2048x1536	2560x1920
IPS	8.76	2.00	0.69	0.52	0.32	0.21	0.14
Efficiency	64.32	89.94	91.54	93.47	98.48	98.36	97.91

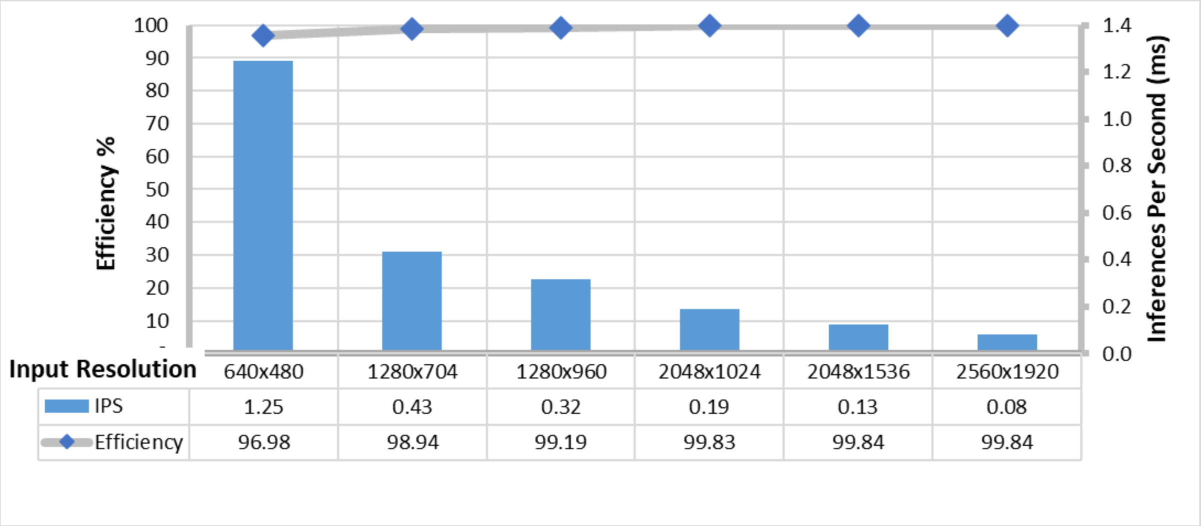
5.10 VGG16

The vgg16 and vgg19 workloads are widely benchmarked public NNs used in the ImageNet benchmark, as well as being used as a base network for other use cases. It was developed with simple 3x3 convolutions, and was one of the first successful CNNs developed. This network is more computationally demanding than its modern counterparts. We use its base network without the fully connected layers a, as ImageNet classification is not a relevant use case in automotive.



Input Resolution	640x480	1280x704	1280x960	2048x1024	2048x1536	2560x1920	3840x2176
IPS	1.59	0.55	0.41	0.24	0.16	0.10	0.06
Efficiency	97.18	98.97	99.20	99.82	99.82	99.83	99.81

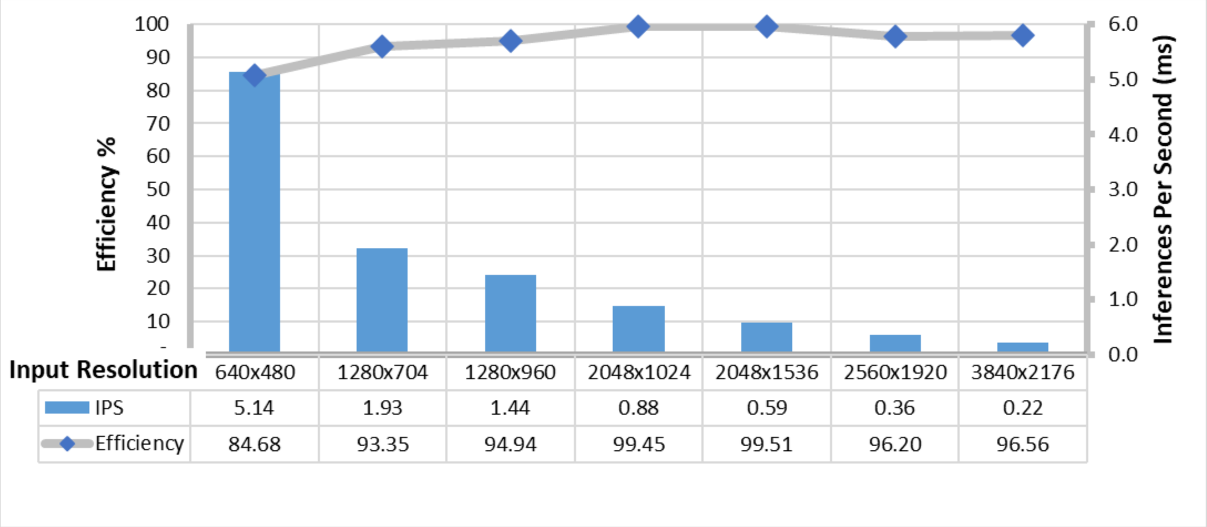
5.11 VGG19



Input Resolution	640x480	1280x704	1280x960	2048x1024	2048x1536	2560x1920
IPS	1.25	0.43	0.32	0.19	0.13	0.08
Efficiency	96.98	98.94	99.19	99.83	99.84	99.84

5.12 YOLO V2

Yolo_v2 is a successful custom NN developed for bounding box object detection.



Input Resolution	640x480	1280x704	1280x960	2048x1024	2048x1536	2560x1920	3840x2176
IPS	5.14	1.93	1.44	0.88	0.59	0.36	0.22
Efficiency	84.68	93.35	94.94	99.45	99.51	96.20	96.56

6 Appendix A – References

The NN descriptor files used in this benchmark will be made available in both `.nnef` and `.prototxt` (where appropriate) formats. They can be downloaded from **https://aimotive.com/benchmarks/**.

7 APPENDIX B – TERMS AND ABBREVIATIONS

The following terms and abbreviations are used in this document:

Terms	Description
ASIC	Application-Specific Integrated Circuit
NN	Neural Network
FPGA	Field-Programmable Gate Array
GFLOPs	Giga Floating Point Operations
GMACs	Giga Multiply-Accumulate Operations
GOPs	Giga Operations
GPU	Graphics Processing Unit
IPS	Inferences Per Second (sometimes referred to as frames per second)
TOPS	Tera (10^9) Operations per Second (INT8; each Multiplication or Addition counts as one Operation; convolution operations only)
TMACs	Tera (10^9) Multiply and Accumulate Operations per Second (INT8; convolution operations only)

8 Appendix C – Document history

Version	Date	Modified by	Description of the modification
v1.0	04/10/2017	Tamas Forgacs Andras Juhasz	Initial public version
v1.1	04/12/2017	Tamas Forgacs Andras Juhasz	Minor corrections
v1.2	12/12/2017	Tamas Forgacs Andras Juhasz	Measurements extended with new NNs
v1.3	01/02/2018	Tamas Forgacs Andras Juhasz	Measurements extended with new NNS
v1.4	24/01/2019	Peter Frank	Measurements extended with new NNS
v1.45	25/11/2019	Peter Frank Zsolt Ambrus	Bug in MobileNet benchmark framework; results temporarily removed Minor editorial corrections
v2.0	04/03/2020	Peter Frank Marton Feher Tony King-Smith	Initial update for aiWare3P results; change of format
v2.1	11/04/2020	Peter Frank Tony King-Smith	Update following internal review of results
v2.2	19/03/2020	Peter Frank Tony King-Smith	Updates to Measurement Methodology descriptions
v3.0	21/04/2021	Peter Frank Marton Feher Tony King-Smith	Rewrite using latest automated benchmarking hardware platforms and methodologies, and availability of improved hardware platforms for detailed measurements (Apache5 @ 500MHz; larger FPGA)
V3.01	27/04/21	Tony King-Smith	Updated legal Disclaimer; minor typo corrections

Disclaimer

The information in this document is subject to change without notice and describes only the product defined in the introduction of this documentation.

This documentation is intended for the sole purpose of providing information to the reader. No part of it may be used, reproduced, modified or transmitted in any form or means without the prior written permission of AImotive.

The documentation has been prepared to be used by professional and properly trained personnel.
AImotive welcomes comments as part of the process of continuous development and improvement of the documentation.

The information or statements given in this documentation concerning the suitability, capacity, or performance of the mentioned software products are given "as is".

The information disclosed shall not constitute any representation, warranty, assurance, or guarantee by AImotive to the reader of any kind, in particular with respect to the non-infringement of trademarks, patents, copyrights or any other intellectual property rights, or other rights of third parties. However, AImotive has made all reasonable efforts to ensure that the instructions contained in the document are adequate and free of material errors and omissions.

AImotive will, if deemed necessary by AImotive, explain issues which may not be covered by the document.

AImotive will correct errors in this documentation as soon as possible.
IN NO EVENT WILL AIMOTIVE BE LIABLE FOR ERRORS IN THIS DOCUMENTATION OR FOR ANY DAMAGES, INCLUDING BUT NOT LIMITED TO SPECIAL, DIRECT, INDIRECT, INCIDENTAL OR CONSEQUENTIAL OR ANY LOSSES, SUCH AS BUT NOT LIMITED TO LOSS OF PROFIT, REVENUE, BUSINESS INTERRUPTION, BUSINESS OPPORTUNITY OR DATA, THAT MAY ARISE FROM THE USE OF THIS DOCUMENT OR THE INFORMATION IN IT.

This documentation and the product(s) it describes are considered protected by copyrights and other intellectual property rights according to the applicable laws.

Created on	16/03/2021	Author	aiWare Engineering
Last modified	27/04/2021	Version	v3.0
Confidentiality:	Public	Document type	Technical Report

All logos are trademarks of AImotive Kft. Other product names mentioned in this document may be trademarks of their respective owners, and they are mentioned for identification purposes only.

If any provision or part of a provision of this disclaimer shall be or is found by any court of competent jurisdiction or public authority to be invalid or unenforceable, such invalidity or unenforceability shall not affect the other provisions or parts of such provisions of this disclaimer; all of which shall remain in full force and effect.