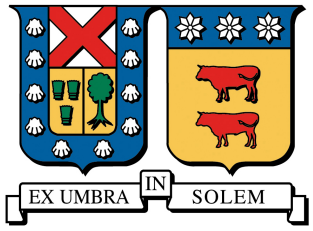
Universidad Técnica Federico Santa María
Departamento de Informática



Feature-Fusion Neck Model for Content-Based Histopathological Image Retrieval

Camilo Esteban Núñez Fernández

Tesis para optar al Grado de
Magíster en Ciencias de la Ingeniería Informática

Valparaiso, Chile
July 24th, 2024

TITULO DE LA TESIS:

Feature-Fusion Neck Model for Content-Based Histopathological Image Retrieval

AUTOR:

Camilo Esteban Núñez Fernández

TRABAJO DE GRADO, presentado en cumplimiento parcial de los requisitos para el Grado de Magíster en Ingeniería Informática de la Universidad Técnica Federico Santa María.

Ph.D. Mauricio Solar _____

Ph.D. Ricardo Ñanculef _____

Ph.D. Rodrigo Salas _____

Valparaíso, Chile
Julio, 2024

Abstract

Feature descriptors in histopathological images pose a significant challenge for the implementation of Content-Based Image Retrieval (CBIR) systems, which are essential tools for assisting pathologists. The complexity arises from the diverse types of tissues and the high dimensionality of Whole Slide Images. Deep learning models like Convolutional Neural Networks and Vision Transformers improve the extraction of these feature descriptors. These models typically generate embeddings by leveraging deeper single-scale linear layers or advanced pooling layers. However, these embeddings, by focusing on local spatial details at a single scale, miss out on the richer spatial context from earlier layers. This gap, pointing towards the development of methods that incorporate multi-scale information to enhance the depth and utility of feature descriptors in histopathological image analysis. In this work, we propose the Local-Global Feature Fusion Embedding Model, an approach composed of a pre-trained backbone for feature extraction from multi-scales, a neck branch for local-global feature fusion, and a Generalized Mean (GeM)-based pooling head for feature descriptors. Based on our experiments, the model's neck and head were trained on ImageNet-1k and PanNuke datasets employing the Sub-center ArcFace loss and compared with the state-of-the-art Kimia Path24C dataset for histopathological image retrieval, achieving a Recall@1 of 99.40% for test patches.

Keywords: Histopathological Image, Content-Based Image Retrieval, Feature Fusion, Feature Embedding, Transfer Learning, Object Detection, Instance Segmentation, Feature Fusion, Context Feature.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The increase in the amount of data produced by healthcare institutions, due to the accessibility and advances in device development, such as whole slide scanners, poses multiple challenges for medical professionals when producing accurate and fast diagnostics [53]. In the histopathology domain, these challenges can be interpreted as an analysis of various types of whole slide images (WSI) that can reach up to $100,000 \times 100,000$ pixels, where tumors can be localized in restricted zones of just a few hundred pixels [28]. However, with the rise of computer-aided approaches in computer vision, specialists have utilized multiple techniques to support their tasks. One such technique is Content-Based Image Retrieval (CBIR), which can assist in the fast and efficient analysis of medical images when compared to hand-crafted analysis. Given the large-scale images of WSI, the use of CBIR techniques in their analysis has become more frequent [38].

In histopathology image analysis, Content-Based Histopathological Image Retrieval (CBHIR) presents a more complex challenge compared to classical CBIR due to the high variability in the visual appearance components of cells in different tissues, including shape, color, or texture [1]. Nevertheless, deep learning models, such as Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs), have greatly contributed to the extraction of these components using pre-trained models [58] on a large-scale dataset like ImageNet [12]. Although these methods can help as a feature extractor for CBHIR systems, they are not specialised enough to be used if the histopathological image domain changes due to the single

modality these pre-trained models have [32]. Because of this, it is often necessary to retrain these models using the transfer-learning strategy to specialize them in the domain of histopathological images. This could be costly in terms of time and computational resources, considering the large number of parameters these pre-trained models have and the large number of patches that public datasets usually have. Even if the domain-specific training is surpassed, the image descriptor is a single-scale embedding that only leverages the deeper single-scale linear layers or advanced pooling layers of the model, missing out on the richer spatial context from earlier layers that a multi-scale perspective of the image can offer.

1.1 Problem Definition

The introduction of Features Pyramid Networks (FPN) in computer vision has sparked a significant shift towards leveraging comprehensive pre-trained networks for feature extraction [43]. Central to this evolution is the increasing adoption of feature fusion within the network's 'neck' segment. This trend capitalizes on the neck's capability to integrate low-level and high-level features from the multi-scale backbone network.

This advancement has led to a proliferation of architectural designs aimed at refining and extending FPN's foundational principles and mechanisms. Notable examples include PANet [44], NAS-FPN [20], and BiFPN [64], highlighting the seminal impact of FPN in promoting multi-level feature fusion.

In the realm of Content-Based Image Retrieval (CBIR), there is a growing interest in multi-scale feature fusion driven by advancements in CNN techniques [41, 8, 56, 66]. However, these techniques have not been extensively explored in multi-scale models for histopathological images.

This study aims to meticulously design, implement, and evaluate a multi-scale feature fusion model using local-global features. The objective is to enhance complexity and semantic richness beyond current models

within a Content-Based Image Retrieval system tailored specifically for histopathological images.

To achieve this, our work presents two main studies:

1. The initial design, implementation, and evaluation of a domain-generalized model to validate the strategy of feature fusion using local-global features in classical visual tasks such as Object Detection and Instance Segmentation.

2. The comprehensive design, implementation, and evaluation of a domain-specific model for histopathological images. This model utilizes the local-global feature fusion approach to generate visual embedding vectors for a specialized Content-Based Image Retrieval system.

By addressing these objectives, this research aims to contribute to the advancement of feature fusion techniques in the domain of histopathological image analysis, potentially improving retrieval accuracy and semantic understanding in medical image applications.

1.2 Hypothesis

Multiple feature vectors can be generated to enhance the robustness and efficiency of CMBIR systems for histopathological images, leveraging a multi-scale local-global feature fusion approach.

1.3 Objectives

1.3.1 General Objective

The general objective of this research is to design, implement, and evaluate a local-global feature fusion neck architecture to generate visual embedding vectors for a specialized Content-Based Image Retrieval system for histopathological images.

1.3.2 Specific Objectives

The specific objectives of this research are as follows:

1. Identify an optimal neck architecture with capabilities of efficient multi-scale feature fusion.
2. Determine an optimal operator block for local and global feature fusion from different receptive fields.
3. Design a neck architecture integrating the identified operator block.
4. Develop an end-to-end model using a backbone, neck, and the Mask R-CNN head detector.
5. Evaluate the end-to-end model using the BiFPN-based neck and the identified neck architecture.
6. Design an end-to-end model for extracting image descriptor embeddings from histopathological images using multi-scale local-global fused features trained with Sub-center ArcFace loss [13].
7. Validate the model on the state-of-the-art CBHIR dataset Kimia Patch24C [55], demonstrating improved Recall@1 through experiments with the embeddings.

1.4 Work Overview

The organization of this document is systematically outlined as follows:

- **Introduction**: This section introduces the research, elucidates the underlying motivation, and defines the specific problem under investigation. It sets the context for the entire study by emphasizing the research's significance and its pertinence to contemporary discussions.
- **Background**: This section delves into the foundational knowledge that informs the research. It explicates key concepts and principles associated with Convolutional Neural Networks, provides an overview of Instance Segmentation, Object Detection tasks, and Content-Based Image Retrieval system, and sets forth the metrics and explanation methods employed to assess the proposed architecture.

- **Related Work**: This section offers a review of prior research and studies pertinent to the subject matter.
- **Proposal**: This segment presents the proposed architecture, detailing its design and implementation.
- **Results and Discussion**: This section furnishes the research outcomes, specifically focusing on the metrics and explanation method results.
- **Conclusion**: This concluding section encapsulates the principal insights derived from the research, underscores the salience of the findings, and contemplates potential trajectories for future research in this domain.

Chapter 2

Background

2.1 Convolutional Neural Networks

To recap our understanding from Feed Forward Networks, each neuron unit is represented and connected in a continuous chain-based architecture. Each layer is described by the following equation:

$$\boldsymbol{h} = f\left(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}\right) \tag{2.1}$$

where $\boldsymbol{h}$ denotes the output of the layer, f is the activation function, $\boldsymbol{W}$ represents the weight vector, $\boldsymbol{b}$ is the bias vector, and $\boldsymbol{x}$ is the input vector of real values such that $\boldsymbol{x} \in \mathbb{R}^n$. In this formulation, the dimensionality $\mathbb{R}^n$ of $\boldsymbol{x}$ confines the input to a one-dimensional vector. Given this constraint, processing images in computer vision tasks using Feed Forward Networks leads to substantial computational complexity in matrix multiplication and results in vectors with high dimensionality of parameters.

Convolutional Neural Networks, as introduced by *Kunihiko Fukushima* [17], aim to address the challenge of dimensionality by utilizing a grid-like topology [22]. *Le Cun et al.* further developed and elaborated on CNNs in their research [39, 40]. They defined a Convolutional Neural Network based on three pivotal concepts: (I) local receptive fields, (II) shared weights, and (III) spatial or temporal subsampling [40].

2.1.1 Local Receptive Fields

As mentioned in the previous section, Feed Forward Networks consist of fully connected layers. The interaction is facilitated through connections between each neuron in the input layer and the corresponding neuron in the output layer, as illustrated in Figure 2.1.

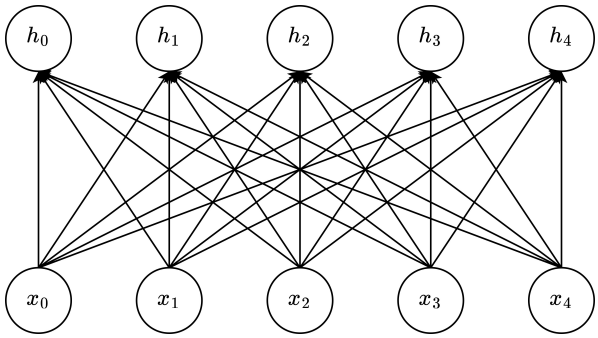


Figure 2.1: Fully connected layer. Given an input $\boldsymbol{x} \in \mathbb{R}^5 : \{x_i \in \boldsymbol{x}, i = 0, 1, ..., 4\}$, the hidden layer $\boldsymbol{h}$ is defined following the equation (2.1).

In CNNs, the connection is often restricted to a small neighborhood within the layer, as depicted in Figure 2.2. This limited neighborhood is termed the 'receptive field' and is employed in the convolution operation to compute sparse interactions in the input layer.

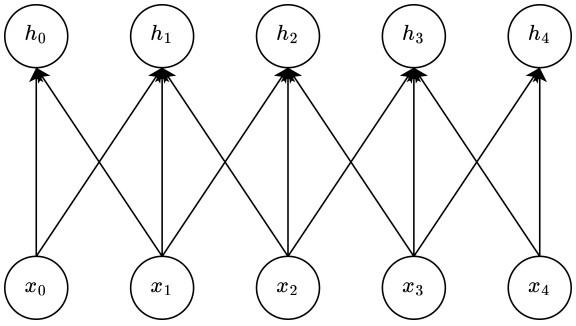


Figure 2.2: Convolutional Layer using neighborhoods: A variation of Figure 2.1

The standard convolutional operation can be represented as:

$$\boldsymbol{S}(t) = (\boldsymbol{x} * \boldsymbol{y})(t)$$

However, this function, being continuous in t, is not directly applicable in a discrete setting. To address this, a discrete form of the convolution

operation can be adopted [4]:

$$\boldsymbol{S}[t] = \boldsymbol{x}[t] * \boldsymbol{y}[t] = \sum_{k=-\infty}^{\infty} \boldsymbol{x}[k] \cdot \boldsymbol{y}[n-k]. \tag{2.2}$$

Here, Equation (2.2) alludes to a 1D convolutional operation, indicating that both $\boldsymbol{x}$ and $\boldsymbol{y}$ are 1D. For 2D operations [4], the equation takes the form:

$$\boldsymbol{S}[m,n] = \boldsymbol{x}[m,n] * \boldsymbol{y}[m,n] = \sum_{j=-\infty}^{\infty} \sum_{i=-\infty}^{\infty} \boldsymbol{x}[i,j] \cdot \boldsymbol{y}[m-i, n-j] \tag{2.3}$$

Equation (2.3) correlates directly with the receptive field concept: in this context, $\boldsymbol{x}$ represents the input, $\boldsymbol{y}$ is the kernel, and $\boldsymbol{S}$ denotes the feature map. Within this framework, connections remain localized within the 2D space of both the input vector and the kernel, as demonstrated in Figure 2.3.

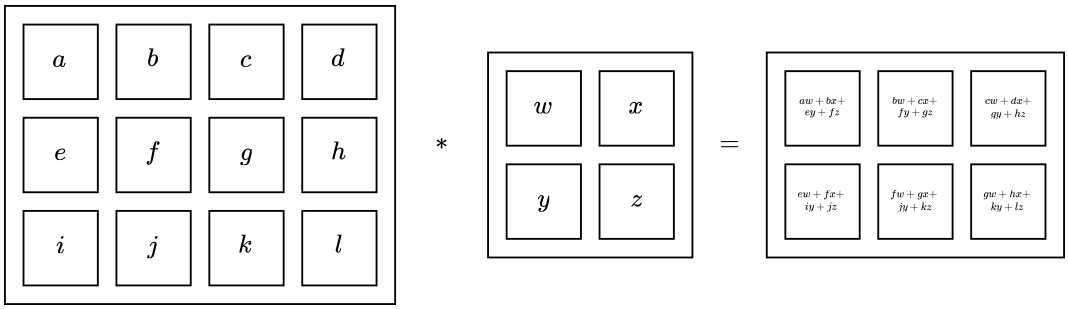


Figure 2.3: Example of a convolutional operation involving an input $\boldsymbol{x}$ and a kernel $\boldsymbol{y}$. Adapted image sourced from [22].

2.1.2 Shared Weights

In CNNs, the convolutional layers use the equation (2.3) to operate over the two dimension. In this process, the kernel $\boldsymbol{y}$ is shared for more than one function in a model [22]. This shared kernel is also know as the filter of the convolution that is convolved with the input. The Figure 2.3, display the a simple plane as input within which all the units share the same set of weights [40]. An example of this can see in the Figure 2.4.

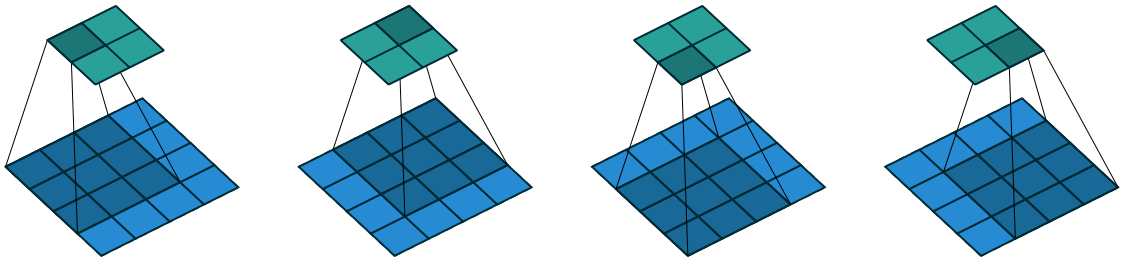


Figure 2.4: Convolving a 3×3 kernel over a 4×4 input using unit strides. Image and caption sourced from [15].

Convolutional Layer

Having defined the concepts of **Local Receptive Fields** and **Shared Weights**, we can express the convolutional layer in terms of equation (2.1) as:

$$\boldsymbol{h} = f\left(\boldsymbol{K} * \boldsymbol{x} + \boldsymbol{b}\right), \tag{2.4}$$

where the operator * represents the convolution between the kernel vector $\boldsymbol{K}$ (or the shared weights) and the input vector $\boldsymbol{x}$.

2.1.3 Spatial or Temporal Subsampling

After applying the convolution operation, the positions of the values in the input vector become less relevant in the output vector or feature map. This can be problematic, especially as these positions may vary across different instances [40]. This issue becomes even more pronounced when multiple convolutions are applied, as commonly occurs in layers. As a result, the extraction of distinctive features might shift in the location of the receptive field. One approach to address this problem is by reducing the precision with which the positions of distinctive features are encoded in a feature map. This reduction is akin to subsampling the output layer and is typically executed after one or more convolutional layers.

Pooling Operation

One operation used to reduce the feature map's dimensionality is the *Pooling* operation, typically applied after one or more convolutional layers. This

operation modifies the layer's output by summarizing its values. Consequently, the spatial dimension is reduced, yielding a new representation of the output with fewer parameters and computations.

The two most common pooling operations are:

- *Max Pooling* [79]: This operation selects the maximum value from defined patches of a feature map.
- *Average Pooling* [40]: This operation computes the average value over specified patches of a feature map.

Both operations result in a downsampled version of the output feature vector, helping to produce a representation that remains relatively invariant to minor shifts in the input vector [22].

2.1.4 Computer Vision Tasks: Instance Segmentation and Object Detection

Convolutional Neural Networks have been instrumental in the realm of computer vision, initially applied to image classification tasks [39, 40]. In these tasks, the network classifies images based on elementary visual features extracted, such as corners, edges, and endpoints. These features, processed across multiple layers of the network, are optimized to best represent the image in question.

Emerging from this foundational task, the versatility of CNN-based approaches has expanded to encompass a broader spectrum of tasks like object detection, semantic segmentation, instance segmentation, object tracking, and more [82, 76]. The purpose of this work focuses on two critical tasks: Object Detection and Instance Segmentation.

Object Detection

Object detection is an advancement of the primary task of image classification. While image classification identifies the main subject of the image, object detection aims to classify and recognize multiple objects within the image. These individual objects are referred to as *instances*. The objective of object detection is to design a network that detects all of these

instances annotated in an image. In this process, the network, also known as a *detector*, must identify the position of each instance and then assign a coordinate *bounding box* around it.

The foundation of the object detection task wasn't originally built upon the CNN architecture. It began with the pioneering work of *Viola and Jones* [68, 67], who employed various techniques such as Haar-like features, integral images, Adaboost, and cascading classifiers [76]. Subsequently, *Dalal and Triggs* [11] introduced the *Histogram of Oriented Gradients (HOG)* as a feature descriptor. Their detection approach utilized a grid-based decomposition of the image and created a gradient histogram for each grid element. These elements were then classified using a linear SVM.

However, prior models was based in the handcrafted features of the annotations. This problem was solved with the usage of the deep learning models, in more specific, the CNN models, due the ability to extract the high level features of the image. It's possible to identify two type of the CNN-based dectetor: **one-stage and two-stage detector**. In firs case, the two-stage detector, the model use a combination of backbone (in sometimes plus a neck) to extract the high level features and then pass it to a head that do the classify task. In the other hand, the one-stage network use an entire model to extract the feature and do the classify of the object in one pass-through stage.

In this work, we will focus on the two-stage model, which will be further described in Chapter 3.

Instance Segmentation

Much like the previously discussed tasks of classification and object detection, image segmentation leverages the high-level features of an image to classify its pixels based on annotations. This overarching task can be further subdivided, with the most prominent and those explored in this work being semantic segmentation and instance segmentation [48, 23]. Semantic

segmentation classifies each pixel of the image into object categories such as cars, people, or traffic signs. Conversely, instance segmentation delves deeper, identifying and segmenting pixels corresponding to individual instances present in the image, distinguishing between each person, each car, or each traffic sign.

The concept of instance segmentation was pioneered by *Hariharan et al.* in their pioneer work titled *Simultaneous Detection and Segmentation* [25]. Here, a two-stage network was utilized for both object detection and instance segmentation. Following this groundbreaking work, several state-of-the-art instance segmentation architectures emerged, utilizing the two-stage approach. One such architecture will be elaborated upon in Chapter 3.

2.1.5 Metrics and CNN Explanation

To evaluated the performance of the models, it is necessary to summarise the principal metric commonly used for Object Detection and Instance Segmentation. On the other hand, in addition to performance metric of the models, we will define a strategy to evaluate the confidence of the proposal models and their respective variations, using visual and numerical evidence at the moment to compere these different models with each others.

Metrics for Object Detection and Instance Segmentation

Both Object Detection and Instance Segmentation metrics are based in the same metric reasoning: **where** and **what** a correct or incorrect area or object the model predicted. The most common way to measure the **where part**, is using the **Intersection over Union (IoU)**, or the **Jaccard Index** [33, 34], and is define as the measure of the area of intersection between the predicted segmentation mask A (or the predicted bound box in the Object Detection) and the ground truth map B (or the ground truth bound box in the Object Detection), divided by the area of the union between them, and is defined as:

$$\text{IoU} = J(A, B) = \frac{||A \cap B||}{||A \cup B||}, \tag{2.5}$$

where $||\cdot||$ detonated the area of the set, and $J(A, B)$ the **Jaccard Index**, that is also in the range 0 and 1.

The most common way to measure the ***what part*** a correct or incorrect area or object the model predicted from the beginning above definition, is using the classical terms *True positive (TP), False positive (FP) and False negative (FN).* Also, the *True Negative (TN)* result are not used in the most command metrics, because indicate anywhere position where the model do not predict a mask or object and their respective annotation did not provide a mask or object; then the TN are vast and unnecessary to calculate. Whit this concepts, it is possible to define for each class, the **Precision** and **Recall** measure as:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{2.6}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{2.7}$$

Whit this two measure, it is possible to define the **Precision-Recall (PR) curve**, where it is show the relationship between Recall on the x-axis and Precision on the y-axis. In this case, the simple form to define a discrete definitions of this is using the **Intersection over Union** as a threshold for the points in the Precision-Recall (PR) curve:

$$PR(A, B) = \sum_{i=0}^{|A|} \sum_{j=0}^{|B|} PR(A_i, B_j) \cdot I\left[\text{IoU}(A_i, B_i) > \alpha\right], \tag{2.8}$$

where $|\cdot|$ detonated the length of the set, $PR(A_i, B_j$ represent the point of the Precision-Recall (PR) curve for each element of the set A and B, $I\,[]$ the indicator function, and α the IoU threshold. Then, it's possible to define the **mean Average Precision (AP)** as the area under the Precision-Recall (PR) curve for each N classes as:

$$\text{mAP} = \frac{1}{N} \sum_{i=1}^{N} ||PR_i||. \tag{2.9}$$

CNN Model Explanation

In the realm of designing and implementing Convolutional Neural Networks, deeper networks aim to achieve more complex representations of inputs compared to their shallower counterparts [6]. Specifically, for tasks in image processing or computer vision, such intricate representations enable enhanced hierarchical depictions of images [77]. This is attributable to how the multiple layers in the network effectively capture pertinent information. However, these deeper networks often present challenges in interpretability, as their predictions can be difficult to elucidate due to limited insight into their internal operations [35].

To address this challenge, it is possible to employ advanced methods that offer improved interpretability and explanations of model performance. This facilitates a more robust evaluation when comparing different models. In this study, we aim to provide clearer insights into the behaviors of the proposed and tested models by utilizing a visual ***class-discriminative location technique*** and a novel *Remove and Debias* metric technique.

Explanation model using Ablation CAM

Visual explanations detailing how a model captures features across its layers can offer valuable insights into the model's internal behaviors. A novel avenue in this domain is the suite of **Class Activation Mapping (CAM)** based methods [78, 54, 9, 16, 69, 14]. These techniques produce a **visual explanation map** by retaining spatial information throughout convolutional layers and underscoring the significance of each neuron for specific decision regions.

Drawing inspiration from the pioneering work of *Bolei Zhou et al.* [78], the **visual explanation map**, given a model prediction f for a target class c of an input image x, can be formulated as:

$$L^c_{\text{CAM}}(A) = \text{ReLU}\left(\sum_{k=1}^{N_l} w^c_k A_k\right), \tag{2.10}$$

where $A = f^{[l]}(x)$ denotes the output of the l-th layer, A_k represents the

k-th activation map of A, and w_k^c is the k-th weight corresponding to class c. The symbol N_l designates the count of activation maps in the l-th layer. Within the CAM paradigm, the coefficient w_k^c indicates the significance of A_k [78]. Applying the ReLU activation to the linear combination filters only the positive map values, thereby pinpointing pixels influencing the target class c.

In this research, we adopt the approach described by *Saurabh Desai et al.* in **Ablation CAM** [14]. This method is favored to avoid the gradient saturation issues faced by gradient-weighted strategies like *Grad-CAM* [54]. Gradient saturation in these techniques leads to diminishing backpropagation gradients, consequently causing visualizations to falter in identifying pertinent image regions [14, 16].

Ablation CAM introduces a procedure wherein, for a given input image x, a first forward pass through the model yields a non-linear function termed the **class activation score** y^c for class c, derived from the activation map A_k of the final convolutional layer. In a subsequent forward pass with the same input image x, individual activation cell values of the activation map A_k are set to zero. This leads to the ablation of the k-th unit, generating a new **class activation score** y_k^c to serve as a baseline for the activation map A_k. The slope produced by this unit's ablation is defined as:

$$\text{slope } = \frac{y^c - y_k^c}{\|A_k\|}. \tag{2.11}$$

Ablation CAM utilizes a modified version of this *slope* to avoid small values that arise when the norm $\|A_k\|$ is significantly larger than the difference $y^c - y_k^c$. This modification defines the coefficient w_k^c in equation (2.10) as:

$$w_k^c = \frac{y^c - y_k^c}{y^c}, \tag{2.12}$$

and this can be interpreted as the relative decrease in class activation score c upon the removal of activation map A_k [14].

2.2 Content-Based Image Retrieval

Content-Based Image Retrieval (CBIR) is a technique for retrieving relevant images from a large database based on the content of the images themselves, rather than relying on metadata such as tags, keywords, or descriptions. CBIR systems are designed to index and retrieve images by analyzing their visual content, including features like color, texture, shape, and spatial relationships.

2.2.1 Architecture of a CBIR

Based in the flow diagram of the Figure 2.5, the architecture of a Content-Based Image Retrieval system includes several key components and their interactions. The classical definition of each part as shown in the Figure are:

- Images/Embeddings: This component represents the database of images. It can either store the raw images or their embeddings, which are compact feature representations extracted from the images.
- Feature Extraction: This component is responsible for processing the input images or embeddings to extract meaningful features. These features can include color histograms, texture descriptors, shape features, or more complex embeddings obtained through deep learning models.
- Search Index Motor: The search index motor organizes and indexes the extracted features to facilitate efficient retrieval. It uses indexing structures like k-d trees, R-trees, or other advanced methods to manage the feature space.
- Similarity Measurement: This component measures the similarity between the query image's features and the features of the indexed images. It computes distance metrics like Euclidean distance, cosine similarity, or other domain-specific measures to rank the retrieved images based on their relevance to the query.

- Retrieval Visualization: The retrieval visualization module handles the presentation of the search results to the user. It visualizes the images that are most similar to the query, typically in a ranked order, and provides an interface for user interaction.

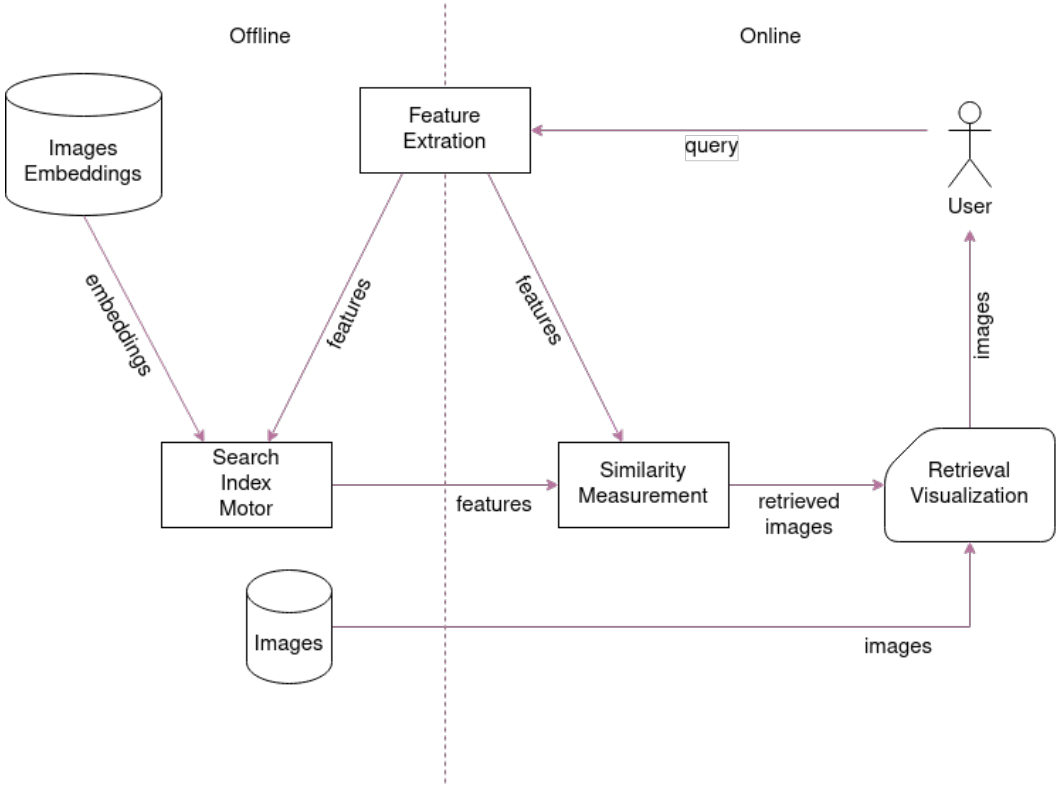


Figure 2.5: Architecture of a Content-Based Image Retrieval (CBIR) System. This diagram illustrates the main components and workflow of a CBIR system. Images from the database are processed through feature extraction, which are then indexed by the search index motor. When a user submits a query, its features are extracted and compared to the indexed features through similarity measurement. The results are visualized and presented to the user, showing the most relevant images retrieved from the database. Image modified from [38].

2.2.2 Metrics CBIR

In Content-Based Image Retrieval systems, several metrics are used to evaluate the performance of the retrieval process. These metrics help quantify how well the system retrieves relevant images in response to a query. Below are some of the common evaluation metrics and the used for this work.

Precision

Precision is the ratio of the number of relevant images retrieved to the total number of images retrieved, and is defined as:

$$\text{Precision} = \frac{|\{\text{Relevant Images}\} \cap \{\text{Retrieved Images}\}|}{|\{\text{Retrieved Images}\}|} \tag{2.13}$$

Recall

Recall is the ratio of the number of relevant images retrieved to the total number of relevant images in the database. Then, the Recall is defined as:

$$\text{Recall} = \frac{|\{\text{Relevant Images}\} \cap \{\text{Retrieved Images}\}|}{|\{\text{Relevant Images}\}|} \tag{2.14}$$

Mean Average Precision (mAP)

Mean Average Precision (mAP) is the mean of the Average Precision (AP) values for a set of queries. AP for a single query is the average of precision values obtained at different levels of recall, and is defined as:

$$\text{AP} = \frac{1}{|\{\text{Relevant Images}\}|} \sum_{k=1}^{n} P(k) \cdot \text{rel}(k) \tag{2.15}$$

where $P(k)$ is the precision at rank k, $\text{rel}(k)$ is a binary function that is 1 if the image at rank k is relevant and 0 otherwise, and n is the total number of retrieved images. Then given Average Precision definition, the Mean Average Precision is define as:

$$\text{mAP} = \frac{1}{Q} \sum_{q=1}^{Q} \text{AP}_q \tag{2.16}$$

where Q is the total number of queries.

Chapter 3

Related Work

3.1 Feature Fusion Techniques

In this section, we will explore the types of Feature Fusion Techniques represented by the state-of-the-art neck *BiFPN* [64] architecture and theirs precursor architectures *FPN* [43], *PANet* [44], and *NAS-FPN* [20].

3.1.1 FPN

The prior use of featurized image pyramids [2] in object detection was aimed at leveraging their scale-invariance. This unique property allows a model to detect objects over a vast scale range by examining in the pyramid levels. As a result, the detector can generate a multi-scale feature representation where every level, possesses robust semantic value. However, employing this multi-scale feature without an appropriate method of feature fusion can lead to significant semantic discrepancies due to varied depths. To address this challenge, *Tsung-Yi Lin et al.* introduced [43] an architecture that adeptly merges low-resolution features rich in semantics with high-resolution features that are semantically weaker. This is achieved through a top-down pathway combined with lateral connections.

3.1.2 PANet

Building upon the architecture of the Feature Pyramid Network (FPN) [43], *Shu Liu et al.* extended their research [44] to introduce a more sophisticated architecture termed the *Path Aggregation Network (PANet)*. They

pinpointed a significant challenge in the foundational FPN design: a lengthened pathway from the basic structural elements to the most advanced features. This elongated route hinders the extraction of detailed localization information. Notably, accessing features at the foundational levels, which are crucial for pinpointing larger instances, becomes increasingly difficult.

In tackling the challenge, the *Bottom-up Path Augmentation* method is adopted. This method amplifies the entire feature hierarchy's localization ability by highlighting the dominant responses of foundational patterns.

3.1.3 NAS-FPN

In the study presented by *Golnaz Ghiasi et al.*, they introduced a novel architecture termed *NAS-FPN* [20]. This architecture leverages an extensive search space to discern the optimal arrangement of unit merge routes. The technique they adopted is based on the Neural Architecture Search (NAS) methodology [80], which was previously proposed by *Zoph et al.* [81]. A distinctive feature of this algorithm is its capability to craft modular architectures. Such modular structures can be efficiently replicated and stacked, culminating in a scalable design. Inspired by this modular concept, *Golnaz Ghiasi et al.* designed a search space adept at churning out scalable architectures, especially those generating pyramidal representations.

3.1.4 BiFPN

The imperative nature of devising an architectural framework that offers scalability in detection, while simultaneously ensuring enhanced accuracy and superior efficiency across an expansive range of resource constraints, has driven researchers to seek innovative solutions. In light of this, *Mingxing Tan et al.* introduced the model termed *Bi-Directional Feature Pyramid Network (BiFPN)* [64]. This model endeavors to achieve elevated performance metrics in one-stage detectors. It does so by harnessing the capabilities of an efficient multi-scale feature fusion coupled with learnable weights. These weights are strategically employed to discern and adapt to

the significance of varied input features.

In this study, the researcher critically evaluated the *Multi-Scale Feature Representations* as proposed by the NAS-FPN model [20]. It was observed that the execution of this model necessitates a disproportionately high consumption of resources, primarily due to the vastness of its search space. Furthermore, when inspecting architectures such as FPN [43] and NAS-FPN [20], there seems to be a discrepancy in the manner in which different input features are fused, resulting in an inconsistent output feature fusion. To address these shortcomings and augment the model's efficiency, the BiFPN was introduced, which incorporates *cross-scale connections*. These connections are specifically designed to rectify the aforementioned inefficiencies in the previously discussed architectures. The optimizations encapsulated within BiFPN include:

1. The elimination of nodes possessing only a singular input edge. This is orchestrated with the objective of eschewing nodes that do not actively participate in the fusion of different feature maps, thus ensuring that only the significant nodes with multiple connections contribute to feature amalgamation.

2. The integration of an additional edge that extends from the original input feature maps to the corresponding output node, provided they reside on the same hierarchical level. This is pursued to facilitate a more comprehensive feature fusion without necessitating the introduction of a plethora of operators.

3. The incorporation of a bi-directional pathway (both top-down and bottom-up) within a singular feature network layer. This layer is then replicated numerous times to foster an enriched high-level feature fusion process.

A graphic representation detailing these optimizations, showcasing BiFPN as the feature network, can be found in Figure 3.1.

In the preceding architectures, there is a discernible uniformity in the treatment of input features. This homogeneity often overlooks the nuanced

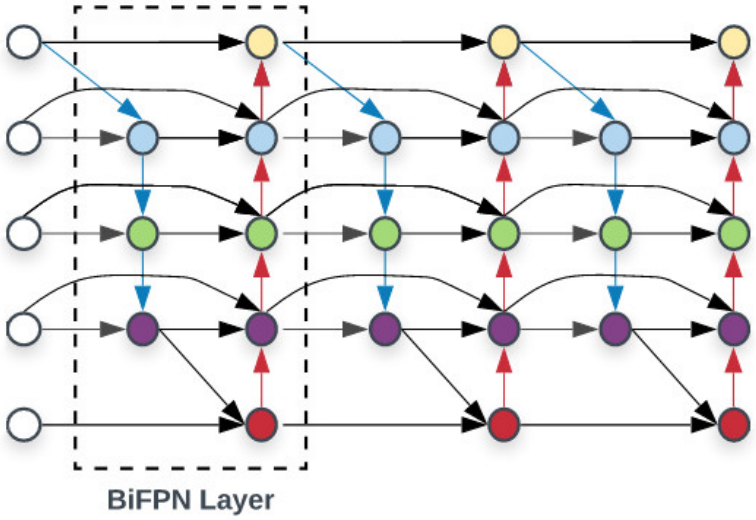


Figure 3.1: Diagram illustrating of BiFPN as the feature network. Image adapted from [72].

significance that some features might possess over others. Contrary to this general approach, BiFPN employs a strategy that allocates a unique weight to each input. This allocation is not arbitrary; rather, it is meticulously designed to facilitate the network's learning of the relative importance of each input feature. This weighting mechanism is encapsulated in the methodology termed *Fast Normalized Fusion*, which is described as:

$$O = \sum_i \frac{w_i}{\epsilon + \sum_j w_j} \cdot I_i, \tag{3.1}$$

In the above equation, I_i delineates the i-th fused feature map. The term w_i represents the weight of the corresponding feature map, and to ensure its non-negativity, a Rectified Linear Unit (ReLU) is applied post determination of each w_i. The constant $\epsilon = 0.0001$ is judiciously incorporated to circumvent potential numerical instabilities that might arise during calculations.

3.2 Histopathological Features Representation for Image Retrieval

In this section, we analyzed the most recent works and proposals related to: (I) Content-Based Histopathological Image Retrieval using deep learning models as feature extractors and (II) a model that is using a strategy of multi-scale receptive fields and local-global feature fusion for the image descriptor in the context of histopathological images.

The actual design of Content-Based Histopathological Image Retrieval is based on deep learning models for the feature extraction process as image descriptors embedding, especially Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs). For example, Hegde et al.[28] propose a similar image search tool named SMILY, based on a CNN ranking model which produces the embedding image descriptor from the tissue samples collection of TCGA. The model used by SMILY was trained with a pair multi-similarity loss using the distance between the embeddings of natural images (e.g., trains, animals, persons, etc)[3].

Another proposal that uses CNN models trained with the multi-similarity loss is proposed by Yang et al.[74], where they used a mixed attention mechanism with spatial attention and channel attention. The model was trained using a self-established histopathological image retrieval dataset and the public dataset Kimia Path24C[55]. In this project, the author specifies that the embedding of the image descriptor is extracted using bottleneck operations composed of *Squeeze-and-Excitation* blocks [30] and fully-connected layers.

Another approach with CNNs is proposed by Tabatabaei et al. in their work [61], where they used a Convolutional Auto-Encoder with unsupervised training. The proposed model reconstructs the input image and extracts the image descriptor embedding from the bottleneck of the auto-encoder.

Alizadeh et al. used a novel siamese CNN hashing model [49] to avoid imbalanced classes and limited samples in histopathological image datasets. The siamese model uses two pre-trained models with shared weights as feature extractors and a hash code generator layer. It was trained using a pair contrastive loss function and the public datasets Kather [36] and BreakHis [60].

A similar approach with a siamese CNN as a feature extractor and trained with a contrastive loss can be found in the work of Tabatabaei et al.[62], where the embeddings of the image descriptor are extracted from the deeper layers using global average pooling. The siamese model was trained for two specific histopathological image domains: skin cancer with a dataset of spitzoid melanocytic skin cancer provided by the University Clinic Hospital of Valencia, and breast cancer with the dataset BreakHis[60].

In the paradigm of multi-scale and local-global feature descriptors, the work of Iqbal et al. [32] presents a novel approach for the fusion of textural features extracted from a Global-Local Pyramid Pattern (GLPP) [31] and visual features extracted with a CNN trained in multiple medical image domains like X-ray, breast tumors, and skin lesions.

Chapter 4

Proposal

In alignment with the objectives delineated in Section 1.3.2, this chapter presents the newly proposed architectural framework termed **Local-Global Feature Fusion Detection Model (LGFFDM)** and their respective implementation for the CBIR task, **Local-Global Feature Fusion Embedding Model (LGFFEM)**.

4.1 LGFFDM: Local-Global Feature Fusion Detection Model

LGFFDM architecture design draws inspiration from the *two-stage detector* paradigm, as comprehensively elucidated in Chapter 3. The architecture is strategically segmented into three primary components:

1. **Backbone**: This component represents the refined network specifically tailored for feature extraction. Within this segment, we harness feature maps from diverse stages of the network. The intent is to meticulously capture feature maps both from advanced (high-level) stages and foundational (low-level) stages of the network.

2. **Neck**: The *neck* merges connections from various stages of the backbone. It incorporates the structure of the *BiFPN* [64]. Within this framework, we introduced two new components: the **Local Aggregator** and the **Global Aggregator**. A deeper exploration of these components will be provided in Section 4.1.1.

3. **Head Detector**: This segment integrates a *Mask R-CNN* head [27], equipped with two distinct *RoI Pooler* operators. While one is dedicated to the Object Detection task, the other is tailored for the Instance Segmentation task. These poolers subsequently receive inputs from the varied fused levels extended by the *neck*.

A comprehensive visualization of the final architecture is presented in Figure 4.1.

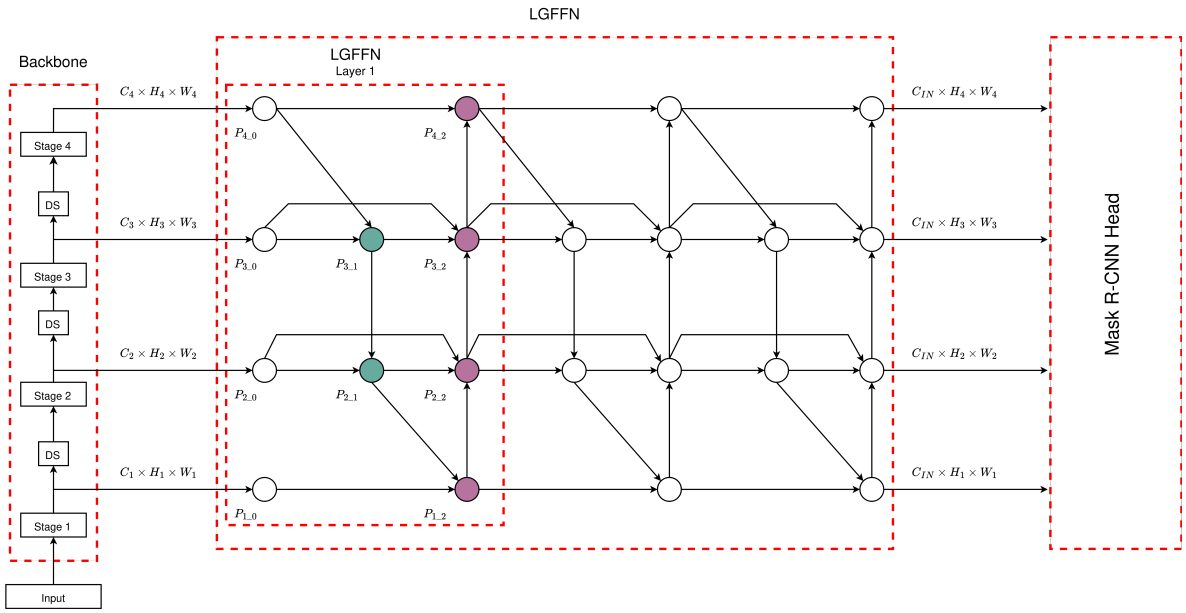


Figure 4.1: Detailed schematic of the LGFFDM architecture.

The subsequent sections will delve into the design intricacies of the *Feature Aggregator Units* located within the *neck*, emphasizing their integration within the *LGFFDM* structure.

4.1.1 Feature Aggregator Units

Drawing parallels with *ParseNet* [45], the *Feature Aggregator Units* are designed to synergize both local and global features emanating from the backbone stages. Within the scope of this research, the distinctions between local and global features are articulated as follows:

1. **Local Feature**: The features given possessing minimal receptive field, these features preserve intricate spatial information, thereby facilitating the generation of high-level features.

2. **Global Feature**: The features extracted from a generalization operation from expansive receptive field, are adept at capturing robust semantic information. These are categorized as low-level features.

Global Operator

In the *Squeeze-and-Excitation* networks study [30], the convolutional vector operation presented in Equation (2.4), denoted as $\boldsymbol{K} * \boldsymbol{x}$, involves a summation across all channels C. The output from this operation is closely interwoven with the local spatial correlations captured by the filters $\boldsymbol{K}$. To address the convolutional complexities, the squeeze-and-excitation block integrates global spatial information throughout the channels and subsequently consolidates this data through channel-wise dependencies. Then, for an input $\mathbf{X} \in \mathbb{R}^{C \times H \times W}$, the *global feature operator* of the squeeze-and-excitation block is defined as follows:

$$\mathbf{G}(\mathbf{X}) = \mathbf{W}_2 \left(\delta \left(\mathbf{W}_1(\mathbf{g}(\mathbf{X})) \right) \right), \tag{4.1}$$

where both $\mathbf{W}_1 \in \mathbb{R}^{\frac{C}{r} \times C}$ and $\mathbf{W}_2 \in \mathbb{R}^{C \times \frac{C}{r}}$ represent learnable weights. The channel reduction factor is denoted as r and usually takes values within the set $\{2, 4\}$. Here, δ signifies the activation function. The squeeze operator, symbolized as $\mathbf{g}(\mathbf{X})$, is formulated as:

$$g(\mathbf{X})_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} x_{c,i,j}. \tag{4.2}$$

In this equation, $g(\mathbf{X})_c$ is a component of $\mathbf{g}(\mathbf{X}) \in \mathbb{R}^C$, and $x_{c,i,j}$ corresponds to individual units of the feature map $\mathbf{X}$.

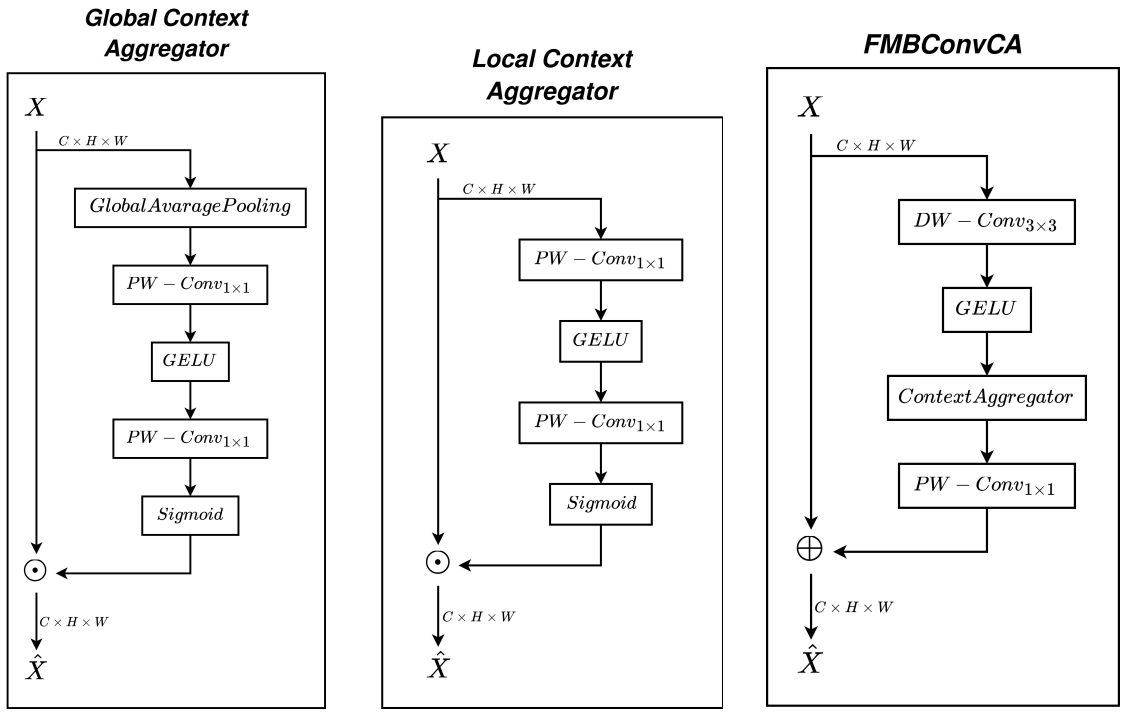
Building upon the concept of point-wise channel representation introduced by the *global feature operator*, we focus on the interaction of the global spatial information throughout the channels. This approach mirrors that of the *MS-CAM* block presented by *Yimian Dai et al.* in their research

titled *Attentional Feature Fusion* [10]. Guided by their insights, we have chosen to implement the *point-wise convolution* for channel aggregation, specifically targeting the learnable weights $\mathbf{W}_1$ and $\mathbf{W}_2$.

As a result, the **Global Operator** formulated in our study can be expressed through the following equation:

$$\sigma\left(\mathbf{G}(\mathbf{X})\right) = \sigma\left(\text{PW-Conv}_{1\times1}\left(\delta\left(\text{PW-Conv}_{1\times1}(\mathbf{g}(\mathbf{X}))\right)\right)\right). \tag{4.3}$$

Within this equation, σ denotes the *Sigmoid* function, while δ refers to the GELU [29] activation function. The term PW-Conv$_{1\times1}$ signifies the point-wise convolution. Following this, a channel-wise multiplication is performed between the output $\sigma\left(\mathbf{G}(\mathbf{X})\right)$ and the input $\mathbf{X}$. A graphical representation of the **Global Operator** is available in Figure 4.2a.



(a) Detailed schematic Global Aggregator Unit.

(b) Detailed schematic Local Aggregator Unit.

(c) Detailed schematic of the merge unit node.

Figure 4.2: Diagram illustrating the Feature Aggregator Units and merge unit node FMBConvCA, proposed for the LGFFDM.

Local Operator

Given the understanding that channel relationships shaped by convolution are inherently implicit and localized, the *Local Operator* functions as a simplified version of equation (4.3). Notably, this operator excludes the *global feature operator*:

$$\sigma\left(\mathbf{L}(\mathbf{X})\right) = \sigma\left(\text{PW-Conv}_{1\times1}\left(\delta\left(\text{PW-Conv}_{1\times1}(\mathbf{X})\right)\right)\right). \tag{4.4}$$

Mirroring the process in equation (4.3), a channel-wise multiplication is carried out between the output $\sigma\left(\mathbf{L}(\mathbf{X})\right)$ and the input $\mathbf{X}$. A graphical illustration of the **Local Operator** can be found in Figure 4.2b.

FMBConvCA Block

After establishing the architecture of the *Aggregator Units*, we incorporate them into a refined version of the *Fused-MBConv* block [24], hereafter referred to as **FMBConvCA**. This modification is analogous to the enhanced block delineated in the study by *Hatamizadeh et al.* [26]. The purpose of this novel block is to imbue the network with advantageous characteristics, such as inductive bias and the capacity to model inter-channel dependencies, particularly at the various stages within the neck of the architecture. The bottleneck equation for the *FMBConvCA* block can be formally defined as follows:

$$\text{FMBConvCA}\left(\mathbf{X}\right) = \text{PW-Conv}_{1\times1}\left(\text{CA}\left(\delta\left(\text{DW-Conv}_{3\times3}\left(\mathbf{X}\right)\right)\right)\right), \tag{4.5}$$

In Equation 4.5, the term $\text{DW-Conv}_{3\times3}$ denotes a 3×3 depth-wise convolution operation [30]. The symbol δ is employed to represent the Gaussian Error Linear Unit (GELU) activation function [29]. Moreover, $\text{PW-Conv}_{1\times1}$ signifies the point-wise convolution operation. Within this equation, CA stands for the *Aggregator Units*, which can take the form of either $\sigma\left(\mathbf{L}(\mathbf{X})\right)$ or $\sigma\left(\mathbf{G}(\mathbf{X})\right)$. Lastly, the output of the $\text{FMBConvCA}\left(\mathbf{X}\right)$ operation is subject to direct summation with the input vector $\mathbf{X}$. A visual representation of the FMBConvCA block is provided in Figure 4.2c.

The incorporation of the *Aggregator Units* into the **FMBConvCA** block provides enhanced capabilities for contextual understanding and feature extraction. By elegantly combining the depth-wise and point-wise convolutions with the *Aggregator Units* and GELU activation function, the block aims to achieve a balance between computational efficiency and representational power. This design fosters a robust architecture capable of handling intricate data patterns and inter-channel dependencies, thereby amplifying the overall performance of the neck.

4.1.2 Architecture of the LGFFN

As alluded to in the introductory section of this chapter, the salient innovations proposed in this work predominantly reside in the *neck* component of the overall neural network architecture. The schematic representation of this enhanced *neck*, denoted as *Local-Global Feature Fusion Neck (LGFFN)*, is illustrated in Figure 4.1. The proposed architecture incorporates the node structures layer from the *BiFPN* [64] and substitutes the *Fast Normalized Fusion* weighting mechanism with the *FMBConvCA* block within each node.

The input nodes for this specialized neck region are labeled as P_{1_0}, P_{2_0}, P_{3_0}, and P_{4_0}, each of which corresponds to the lateral output from the backend of the network. The intermediate nodes[1], namely P_{3_1} and P_{2_1}, serve as internal aggregation fusion points. Similarly, the terminal nodes[1], denoted as P_{1_2}, P_{2_2}, P_{3_2}, and P_{4_2}, represent the ultimate aggregation fusion nodes and are analogous in function to their counterparts in the traditional *BiFPN* architecture layer.

The formal definitions for each of these feature fusion nodes are enu-

[1]The colors JungleGreen and DarkOrchid are consistent with the color scheme used to denote intermediate and terminal nodes, respectively, in Figure 4.1.

merated below:

$$P_{3\_1} = \text{FMBConvCA-}\mathbf{g}\left(P_{3\_0}\right) \oplus \text{FMBConvCA-}\mathbf{L}\left(\text{Resize}\left(P_{4\_0}\right)\right) \quad (4.6)$$
$$P_{2\_1} = \text{FMBConvCA-}\mathbf{g}\left(P_{2\_0}\right) \oplus \text{FMBConvCA-}\mathbf{L}\left(\text{Resize}\left(P_{3\_1}\right)\right) \quad (4.7)$$
$$P_{1\_2} = \text{FMBConvCA-}\mathbf{g}\left(P_{1\_0}\right) \oplus \text{FMBConvCA-}\mathbf{L}\left(\text{Resize}\left(P_{2\_1}\right)\right) \quad (4.8)$$
$$P_{2\_2} = \text{FMBConvCA-}\mathbf{L}\left(P_{2\_0}\right) \oplus \text{FMBConvCA-}\mathbf{L}\left(P_{2\_1}\right) \quad (4.9)$$
$$\oplus \text{FMBConvCA-}\mathbf{g}\left(\text{Resize}\left(P_{1\_2}\right)\right) \quad (4.10)$$
$$P_{3\_2} = \text{FMBConvCA-}\mathbf{L}\left(P_{3\_0}\right) \oplus \text{FMBConvCA-}\mathbf{L}\left(P_{3\_1}\right) \quad (4.11)$$
$$\oplus \text{FMBConvCA-}\mathbf{g}\left(\text{Resize}\left(P_{2\_2}\right)\right) \quad (4.12)$$
$$P_{4\_2} = \text{FMBConvCA-}\mathbf{L}\left(P_{4\_0}\right) \oplus \text{FMBConvCA-}\mathbf{g}\left(\text{Resize}\left(P_{3\_2}\right)\right) \quad (4.13)$$

In the above equations, FMBConvCA-**L** and FMBConvCA-**g** represent variants of the *FMBConvCA Block* configured for both *Local Operator* and *Global Operator*, and $\oplus$ indicates a direct summation operation.

The LGFFN neck design presents a nuanced approach to achieving efficient feature fusion and contextual understanding. By embedding the *FMBConvCA Block* within the nodes, the architecture enhances its capabilities for capturing complex relationships and spatial dependencies across channels. These intricacies are efficiently managed, thus contributing to the overall robustness and performance of the neck.

4.2 LGFFEM: Local-Global Feature Fusion Embedding Model

The Local-Global Feature Fusion Embedding Model (LGFFEM) framework's is a modified version of the Local-Global Feature Fusion Detection Model (LGFFDM) proposed for the CBIR task, and their design is composes by the same three principal components of the LGFFDM, but with a modified GeM head that create the image descriptor feature embedding given the fused feature from *neck* using multiples mini-heads composed by trainable poling layers, fully-connected layers and normalization operations. A comprehensive visualization of the final architecture is presented in Figure 4.3.

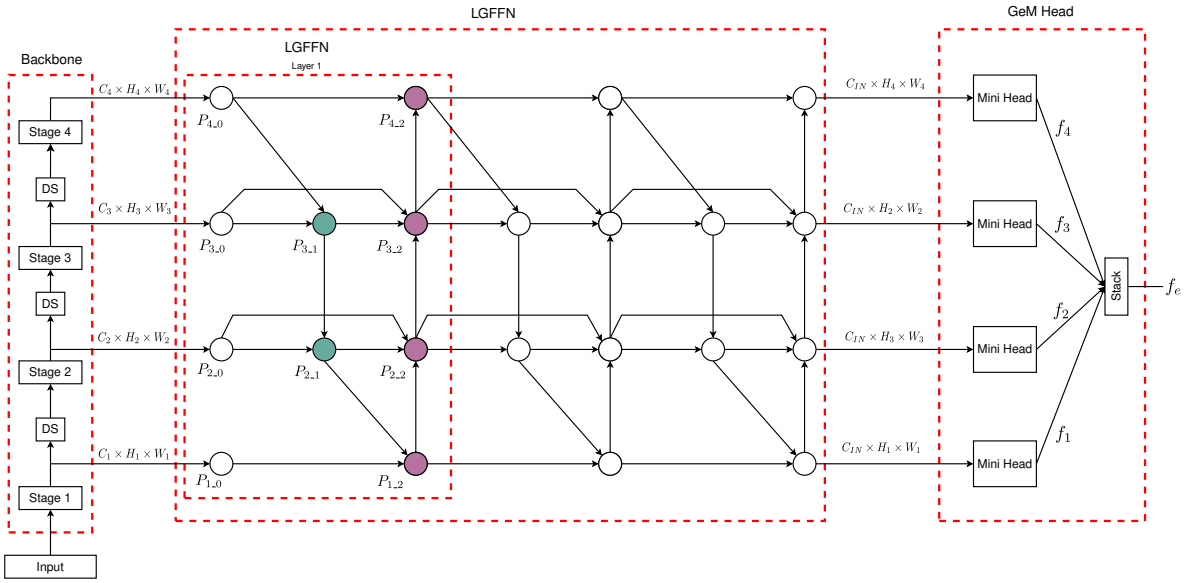


Figure 4.3: Detailed schematic of the LGFFEM architecture. The architecture comprises a pre-trained backbone as a feature extractor from multi-scale stages, a trainable neck consisting of layers for local-global feature fusion, and a pooling head composed of trainable GeM mini-heads for each multi-scale fused feature from the neck.

4.2.1 Modified Feature Aggregator Units

Global Aggregator

The modified *global feature operator* for the LGFFEM architecture, can be formulated through the following equation:

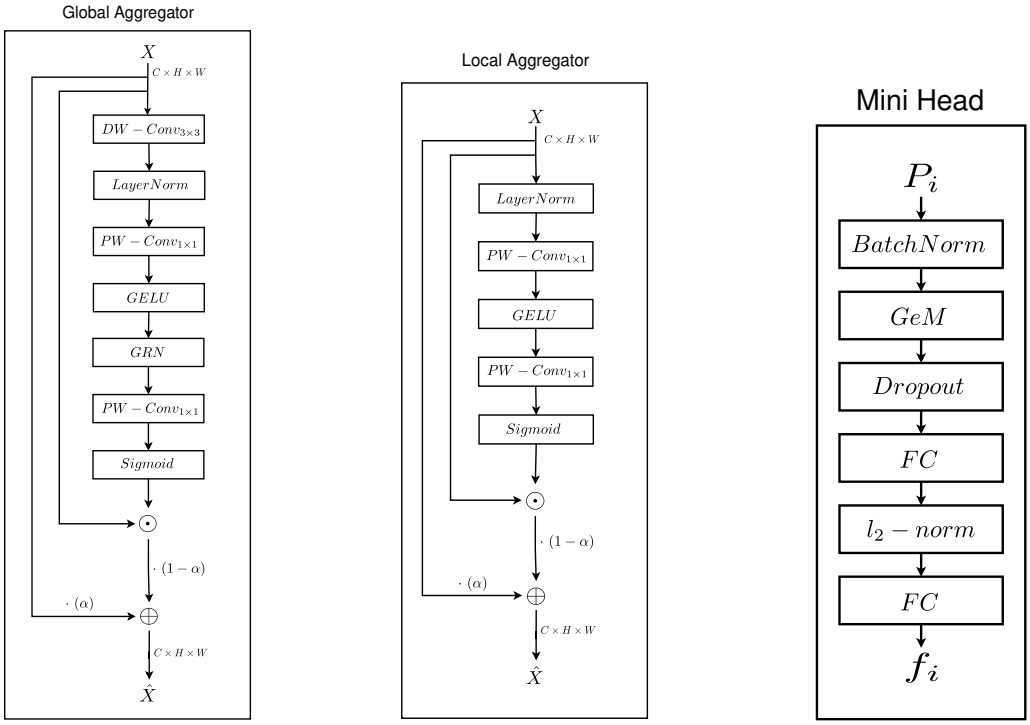
$$\mathbf{G}(\mathbf{X}) = \text{PW-Conv}_{1\times1}\left(\text{GRN}(\delta\left(\text{PW-Conv}_{1\times1}(\text{DW-Conv}_{3\times3}(\mathbf{X}))\right))\right). \tag{4.14}$$

Within this equation, δ refers to the GELU [29] activation function. The term $\text{PW-Conv}_{1\times1}$ signifies the pointwise convolution, and $\text{DW-Conv}_{3\times3}$ signifies the dephwise convolution, while GRN refers to the *Global Response Normalization*(GRN) layer.

As a channel-wise attention, and similar as *Squeeze-and-Excitation* networks, the novel **Global Aggregator** is defined as:

$$\mathbf{GA}(\mathbf{X}) = \alpha \cdot \mathbf{X} \oplus (1-\alpha) \cdot (\mathbf{X} \odot \sigma\left(\mathbf{G}(\mathbf{X})\right)), \tag{4.15}$$

where α is a trainable parameter, and σ denotes the *Sigmoid* function. A graphical representation of the **Global Aggregator** is available in the Figure 4.4a.



(a) Detailed schematic Global Aggregator Unit.

(b) Detailed schematic Local Aggregator Unit.

(c) Detailed schematic Mini Head Unit from GeM Head.

Figure 4.4: Illustration of the bottleneck operation for the Local and Global aggregators and the pooling GeM mini-head

Local Aggregator

Similar as the Local Aggregator of the LGFFDM, the *local feature operator* functions as a simplified version of equation (4.14). Notably, this operator excludes the *generalized spatial dependency operator* $\mathbf{g}(\mathbf{X})$:

$$\mathbf{L}(\mathbf{X}) = \text{PW-Conv}_{1\times1}\left(\delta\left(\text{PW-Conv}_{1\times1}(\mathbf{X})\right)\right). \tag{4.16}$$

Finally, applying the channel-wise attention, the novel **Local Aggregator** is defined as:

$$\mathbf{LA}(\mathbf{X}) = \alpha \cdot \mathbf{X} \oplus (1-\alpha) \cdot (\mathbf{X} \odot \sigma\left(\mathbf{L}(\mathbf{X})\right)), \tag{4.17}$$

A graphical illustration of the **Local Aggregator** can be found in the Figure 4.4b.

4.2.2 Modified LGFFN's Architecture

The schematic representation of the modified LGFFN's architecture is illustrated in Figure 4.3. The formal definitions for each of new feature fusion nodes are enumerated below:

$$P_{3\_1} = \frac{w_1^{3\_1} \cdot \text{GA}\left(P_{3\_0}\right) \oplus w_2^{3\_1} \cdot \text{LA}\left(\text{Resize}\left(P_{4\_0}\right)\right)}{w_1^{3\_1} + w_2^{3\_1} + \epsilon} \tag{4.18}$$

$$P_{2\_1} = \frac{w_1^{2\_1} \cdot \text{GA}\left(P_{2\_0}\right) \oplus w_2^{2\_1} \cdot \text{LA}\left(\text{Resize}\left(P_{3\_1}\right)\right)}{w_1^{2\_1} + w_2^{2\_1} + \epsilon} \tag{4.19}$$

$$P_{1\_2} = \frac{w_1^{1\_2} \cdot \text{GA}\left(P_{1\_0}\right) \oplus w_2^{1\_2} \cdot \text{LA}\left(\text{Resize}\left(P_{2\_1}\right)\right)}{w_1^{1\_2} + w_2^{1\_2} + \epsilon} \tag{4.20}$$

$$P_{2\_2} = \frac{w_1^{2\_2} \cdot \text{LA}\left(P_{2\_0}\right) \oplus w_2^{2\_2} \cdot \text{LA}\left(P_{2\_1}\right) \oplus w_3^{2\_2} \cdot \text{GA}\left(\text{Resize}\left(P_{1\_2}\right)\right)}{w_1^{2\_2} + w_2^{2\_2} + w_3^{2\_2} + \epsilon} \tag{4.21}$$

$$P_{3\_2} = \frac{w_1^{3\_2} \cdot \text{LA}\left(P_{3\_0}\right) \oplus w_2^{3\_2} \cdot \text{LA}\left(P_{3\_1}\right) \oplus w_3^{3\_2} \cdot \text{GA}\left(\text{Resize}\left(P_{2\_2}\right)\right)}{w_1^{3\_2} + w_2^{3\_2} + w_3^{3\_2} + \epsilon} \tag{4.22}$$

$$P_{4\_2} = \frac{w_1^{4\_2} \cdot \text{LA}\left(P_{4\_0}\right) \oplus w_2^{4\_2} \cdot \text{GA}\left(\text{Resize}\left(P_{3\_2}\right)\right)}{w_1^{4\_2} + w_2^{4\_2} + \epsilon} \tag{4.23}$$

In the above equations, LA and GA represent both *Local Aggregator* and *Global Aggregator*, $\oplus$ indicates a direct summation operation, while w_j^i represent the weight for the operation $j \in \{1, 2|1, 2, 3\}$ in the node $i \in \{3\_1, 2\_1, 1\_2, 2\_2, 3\_2, 4\_2\}$, and ϵ a *small value to avoid numerical instability* [64].

4.2.3 Embedding Head

Given the fused 2D features maps from the terminal nodes P_{1_2}, P_{2_2}, P_{3_2}, and P_{4_2} from a layer of the neck, the embedding head will create the image descriptors, as illustrated in the Figure 4.3 like *GeM Head*. This head is composed by four mini heads, one for each terminal nodes, and similar to the works [57, 8], we applied a *Generalized-Mean (GeM)* [52] pooling layer with learnable parameters for the polling process of the 2D feature maps.

Then, given a features map P_i with dimension $C_{in} \times H_i \times W_i$ where $i \in \{1\_2, 2\_2, 3\_2, 4\_2\}$, a mini head produce a vector f_i with size C_{in} produce by:

$$f_i = \mathrm{FC}\left(\mathrm{l}_2\left(\mathrm{FC}\left(\mathrm{GeM}\left(P_i\right)\right)\right)\right), \tag{4.24}$$

where FC represent a fully-connected layer, l_2 the L2 normalization operator, and GeM the Generalized-Mean polling layer. The L2 normalization operator is used in order to applied the *Sub-center ArcFace* [13] loss during the training process and for minimizing the overall loss [70]. This bottleneck operation is illustrated in the supplementary Figure 4.4c.

The final vector image descriptor f_e with size $4 \cdot C_{in}$ is composed by the stacking process of the four feature vectors f_i.

Chapter 5

Results

5.1 Experiments Definitions

The main objective of this study is to thoroughly investigate the effectiveness of our proposed end-to-end neural network models in addressing challenges in two primary areas: (I) object detection and instance segmentation, and (II) content-based image retrieval. To achieve this goal, we have established a meticulous experimental setup to ensure computational rigor and the reproducibility of results.

5.1.1 Hardware Configuration

For the purpose of this study, a specialized hardware configuration was utilized during the experimental phase to facilitate accelerated computation. This hardware was instrumental in training the end-to-end neural network with the assistance of GPU acceleration. A comprehensive list of the hardware components and their respective models is provided in Table 5.1.

Component	**Model**
CPU	*AMD Ryzen 9 5950X 16-Core Processor*
GPU	*NVIDIA GeForce RTX 3090 24GB*
RAM	*64GB*
Data Storage	*NVMe 1TB*

Table 5.1: Detailed hardware configuration employed for model training.

5.1.2 Specifications of Computational Environment

The architectures of the proposed networks was realized through the utilization of the PyTorch framework [50]. To ensure compatibility and leverage optimizations, the computational environment was configured using NVIDIA NGC's official PyTorch container, version 23.05 [1]. This containerized approach facilitates reproducibility and enhances the overall robustness of the experimental setup.

For the quantitative assessment of models performance, metric evaluations were conducted as delineated in Section 2.1.5. Specifically, the PyTorch model for the LGFFDM architecture underwent evaluation via the TorchMetrics API library [2]. This library was chosen for its comprehensive set of metrics that are well-aligned with the evaluation criteria set forth in this study. For the LGFFEM model architecture, the evaluation initially utilized the proposed mean average precision (mAP) implementation introduced in the work *Revisiting Oxford and Paris* [51]. Subsequently, the domain specification metrics for histopathology images were adopted as proposed by the authors in [5], defined as η_p for patch-to-scan accuracy, η_w for whole-scan accuracy, and η_{tot} for total accuracy. As the index retrieval algorithm, the FAISS library [3] was used with an Euclidean index.

Furthermore, the methodologies for CNN explainability were discussed in Section 2.1.5. To elucidate the behavior of the trained models, the CAM strategy employed was Ablation CAM. These strategy were implemented using the *Advanced AI Explainability for PyTorch* package [21].

5.1.3 Training Specifications

LGFFDM Datasets

In order to rigorously assess the performance of our proposed model in the realms of object detection and instance segmentation, we employed the

[1] `https://docs.nvidia.com/deeplearning/frameworks/pytorch-release-notes/rel-23-05.html`

[2] `https://torchmetrics.readthedocs.io/en/stable/`

[3] `https://ai.meta.com/tools/faiss/`

Common Objects in Context (COCO) dataset, as outlined in [42]. Specifically, the 2017 version of this dataset was utilized for both the training and evaluation phases. The COCO dataset is a large-scale, multi-purpose resource that encompasses various tasks within computer vision, including image classification, object detection, semantic segmentation, and instance segmentation. It boasts an extensive collection of 328,000 images, encompassing 2.5 million labeled instances across 91 object classes, 80 of which are designated for the task of instance segmentation.

One salient feature of the COCO dataset that aligns well with the objectives of this research is its proclivity for scale variation and small object instances. A significant portion of object instances in the dataset occupies less than 1% of the overall image area, as noted in [59]. This characteristic makes the COCO dataset an ideal candidate for evaluating how well our proposed model can generalize and perform when confronted with scale variability and diminutive object instances.

Nevertheless, it is imperative to acknowledge the inherent class imbalance within the COCO dataset. Notably, the class labeled `'person'` is disproportionately represented, boasting a far greater number of instances compared to other classes.

LGFFEM Datasets

Our neck model and GeM head for the LGFFEM are trained using three different strategies, utilizing the public training dataset ImageNet-1k [12], which consists of 1,281,167 training images and 1,000 object classes. Additionally, we employed PanNuke [18, 19] dataset with the toolbox PathML[4], containing 189,744 segmented nuclei and encompassing 19 different types of tissues. Additionally, Kimia Patch24C [55], a training dataset consisting of 22,591 training patches from 24 WSIs representing various tissues, was employed. Image augmentation for PanNuke and Kimia training datasets was carried out using the vision tool Albumentations [7].

[4] `http://pathml.org/`

Training Models Configuration

As delineated in Chapter 4, the architectures of the proposed end-to-end models is compartmentalized into three integral components: (I) the backbone, (II) the neck, and (III) the head detector for the LGFFDM architecture; and (I) the backbone for multi-scale feature extraction, (II) the neck and (III) the GeM head image descriptor feature embedding for the LGFFEM architecture.

Backbone Selection and Configuration for LGFFDM architecture For the backbone, we leverage three pre-trained models: InternImage [72], ConvNeXt [46], and EfficientNetV2 [63]. The configurations of these models employed during training are detailed in Table 5.2. The backbones are categorized into three distinct groups, each of which is organized based on the similarity in the number of parameters. Due to constraints in hardware capabilities and the computational time required for training, our experiments focus exclusively on the models belonging to the group labeled as $G1$. However, configuration files for all the proposed groups of backbones are publicly available in the official repository.

Group	Model Name	**# of Parameters (Millions)**
G0	InternImage-T	28M
	ConvNeXt-T	27M
	EfficientNetV2-S	19M
G1	InternImage-S	48M
	ConvNeXt-S	49M
	EfficientNetV2-M	52M
G2	InternImage-B	94M
	ConvNeXt-B	87M
	EfficientNetV2-L	116M

Table 5.2: Configuration groups of the backbones employed in the proposed model, alongside their respective number of parameters in millions.

Each selected backbone model has been pre-trained on an image classification task using the ImageNet dataset [12]. In accordance with established best practices in transfer learning [75], the weights of these models are frozen to serve as feature extractors.

Neck Configuration for LGFFDM architecture The neck component in the training phase adopts two distinct configurations: one based on the original BiFPN model and another based on the proposed LGFFN model. In adherence to the standard configurations utilized in the InternImage work [5], we employ a five-layered architecture for each neck with an output channel size of 256[6].

Head Detector Configuration for LGFFDM architecture As previously stated in the preceding chapter, the head detector employed in the end-to-end model is Mask R-CNN. The configuration for the anchor sizes is set to $(32, 64, 128, 256)$ and aspect ratios are set to $(0.5, 1.0, 1.5, 2.0)$. Both the RoI Pooler Object and the RoI Pooler Mask receive feature outputs labeled as `'P0', 'P1', 'P2', 'P3'` from the neck component.

Backbone Selection LGFFEM architecture The backbone used for the LGFFEM architecture was selected from the best metrics results from the configurations proposed in the Table 5.2. In this case was the pre-trained backbone ConvNeXt V2 [73].

Neck Configuration for LGFFEM architecture Similar as LGFFDM neck's architecture, we only used 3 layers of LGFFN configuration with an inner channel C_{in} size of 512.

GeM embedding head for LGFFEM architecture The GeM polling layer parameters used are $p = 4.6$ and $\epsilon = 1e - 6$. For the GeM head, we used a vector image descriptor f_e size of 2048.

Training Models Procedures

Training procedures for LGFFDM architecture Building upon the aforementioned definitions of backbones and necks, the ultimate configurations utilized for the training implementation in this study comprise the following combinations:

[5] `https://github.com/OpenGVLab/InternImage/tree/master/detection/configs/coco`
[6] C_{IN} in output nodes as depicted in Figure 4.1

- InternImage-S [72] + BiFPN [65]
- ConvNeXt-S [46] + BiFPN [65]
- EfficientNetV2-M [63] + BiFPN [65]
- InternImage-S [72] + LGFFDM (ours)
- ConvNeXt-S [46] + LGFFDM (ours)
- EfficientNetV2-M [63] + LGFFDM (ours)

In alignment with established best practices exemplified by InternImage [72], the backbones components were initialized with weights pretrained on classification tasks. Subsequent training was conducted on the neck components as well as the Mask R-CNN head detector, adhering to a $1x$ schedule comprising 12 epochs[7].

To optimize the networks, the AdamW optimizer [37] was utilized, conforming to the $1x$ schedule. The learning rate was set at 0.001, with a weight decay coefficient of 0.00001. Training was executed with a batch size of two to mitigate the risk of `out-of-memory` errors. It should be noted that, to further avert memory-related complications, the default image size from the ImageNet dataset was employed, as opposed to the larger InternImage dimensions of 1333×800.

Training procedures for LGFFEM architecture We conducted our experiments using five different strategies for the training of the neck and the GeM head: (A) training using only ImageNet-1k, training using ImageNet-1k + PanNuke with a margin hyperparameter of $m = 28.6$ (B.1), $m = m = 17.2$ (B.2), and $m = 5.73$ (B.3), and (C) training using ImageNet-1k + PanNuke + Kimia Patch24C.

All strategies were trained using the Sub-center ArcFace [13], with hyperparameter a scale $s = 64$.

[7]For further insights into common scheduling practices for object detection networks trained from scratch, refer to `https://github.com/facebookresearch/detectron2/blob/main/MODEL_ZOO.md`

The models were optimized using AdamW with a cosine annealing schedule, where the initial learning rate was set at 5e-3, and the minimum learning rate was set at 8e-5. Strategy A used a batch size of 64 for 60 epochs, while strategies A, B.1, B.2, B.3 and C used a batch size of 64 for 300 epochs. All the images were resized to 224×224 pixels.

5.2 LGFFDM Results

5.2.1 Metric Evaluation Results

In accordance with the evaluation conducted on the six trained models delineated in Section 5.1.3, we assessed their performance using the metric of mean average precision (AP) at Intersection over Union (IoU) thresholds of 0.50 and 0.75. Furthermore, a global average performance metric was derived for the entire validation set `val2017` from the COCO dataset. These empirical results are systematically presented in Table 5.3.

Table 5.3: Performance metrics for object detection and instance segmentation on the `val2017` dataset from `COCO2017`. AP denotes the overall mean average precision. AP_{50} and AP_{75} indicate the mean average precision at $IoU = 0.50$ and $IoU = 0.75$ respectively. Meanwhile, AP^{bbox} and AP^{mask} represent the mean average precision for bounding box and mask results, respectively.

	#params		**Mask R-CNN 1x schedule**					
Model	**Neck**	**Backbone**	AP^{bbox}	AP^{bbox}_{50}	AP^{bbox}_{75}	AP^{mask}	AP^{mask}_{50}	AP^{mask}_{75}
InternImage-S [72] + BiFPN [65]	$2.3M$	$48.5M$	0.1654	0.2914	0.1715	0.0341	0.1239	0.0080
ConvNeXt-S [46] + BiFPN [65]	$2.4M$	$49.4M$	0.1884	0.3215	0.1966	0.0405	0.1420	0.0100
EfficientNetV2-M [63] + BiFPN [65]	$2.2M$	$52.2M$	0.2050	0.3507	0.2177	0.0416	0.1531	0.0092
InternImage-S [72] + LGFFDM (ours)	$9.7M$	$48.5M$	0.2071	0.3548	0.2168	0.0421	0.1508	0.0108
ConvNeXt-S [46] + LGFFDM (ours)	$9.8M$	$49.4M$	0.2297	0.3906	0.2416	0.0475	0.1663	0.0117
EfficientNetV2-M [63] + LGFFDM (ours)	$9.6M$	$52.2M$	0.2470	0.4203	0.2592	0.0433	0.1648	0.0083

5.2.2 Ablation CAM Results

Due to the limitations of the measure provided in the previous section, which does not offer a comprehensive understanding of the neck model behavior, we aim to augment our model evaluation. To achieve this, we will examine the Class Activation Mapping (CAM) across five layers of the necks, for each of the six trained models, utilizing the Ablation CAM visual

explanation technique.

To provide a robust assessment, one disparate image from the COCO dataset have been selected at random, identifiable by their respective ID `171382`. This images serve as the ground truth and is accompanied by their respective bounding boxes and segmentation masks, all of which are depicted in Figure 5.1.

Upon selection of this test image, we have conducted predictions to generate bounding boxes and segmentation masks. The results of these predictions are systematically represented in Figures 5.2.

After applying the Ablation CAM visual explanation technique to the five layers of the neck in each trained model, we generated a series of visual results. For image ID `171382`, these are shown in Figures 5.3 to 5.8.

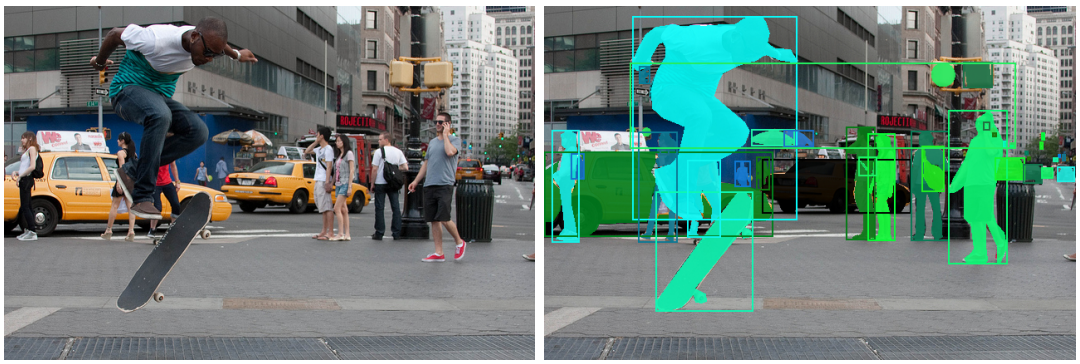
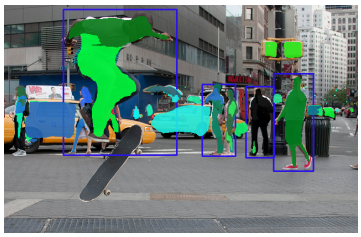
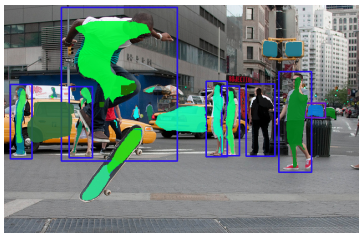


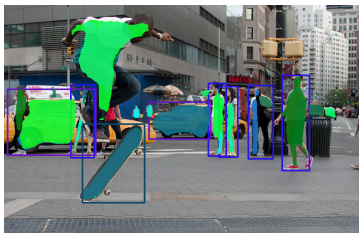
Figure 5.1: Original image from `COCO2017` utilized for testing. The first column displays image ID `171382`. The second column showcases the bounding boxes and instance segmentation masks derived from the annotations of this image.



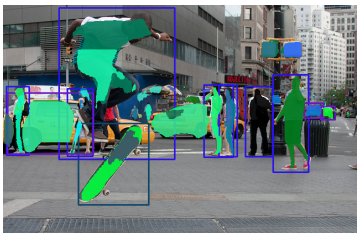
(a) Predictions using InternImage-S + BiFPN.



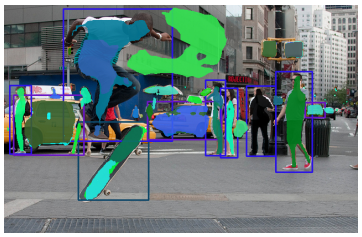
(b) Predictions using ConvNeXt-S + BiFPN.



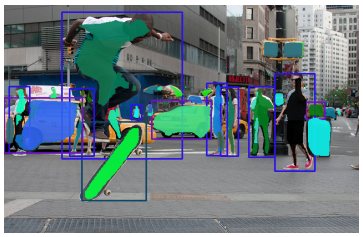
(c) Predictions using EfficientNetV2-M + BiFPN.



(d) Predictions using InternImage-S + LGFFDM.

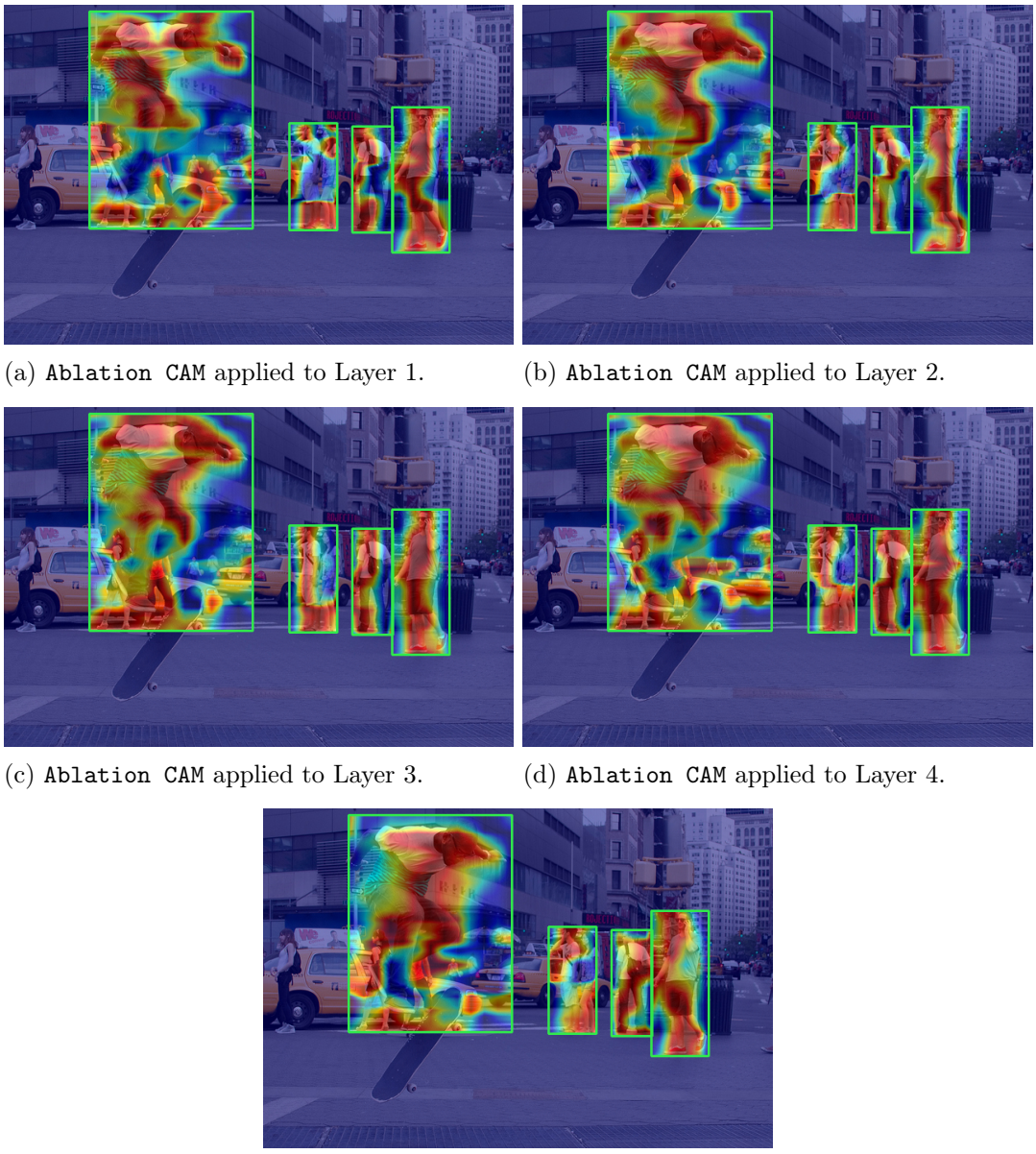


(e) Predictions using ConvNeXt-S + LGFFDM.



(f) Predictions using EfficientNetV2-M + LGFFDM.

Figure 5.2: Object detection and instance segmentation predictions for image `ID 171382` based on evaluations of the six tested models.



(a) `Ablation CAM` applied to Layer 1.

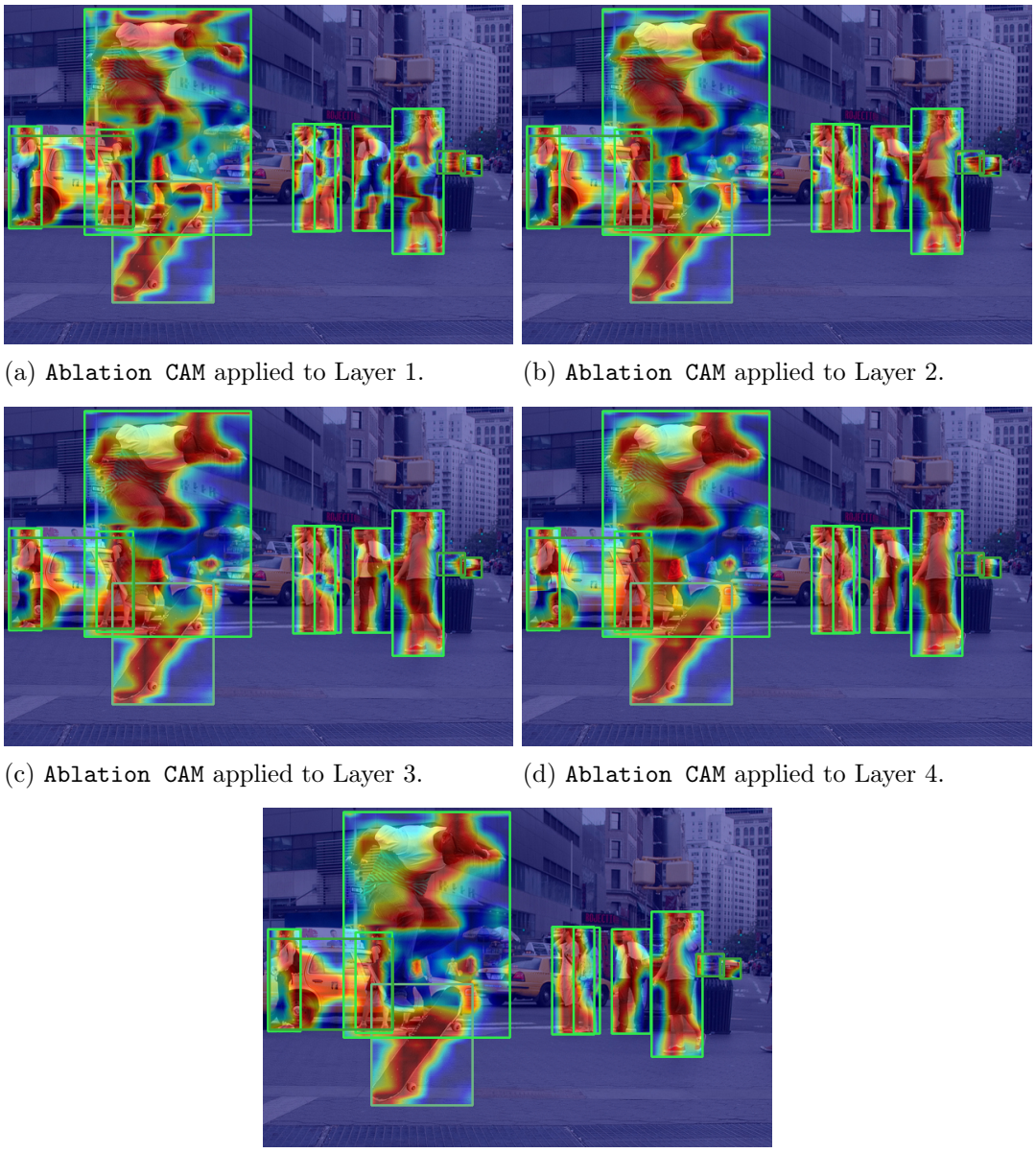
(b) `Ablation CAM` applied to Layer 2.

(c) `Ablation CAM` applied to Layer 3.

(d) `Ablation CAM` applied to Layer 4.

(e) `Ablation CAM` applied to Layer 5.

Figure 5.3: `Ablation CAM` applied to the five layers in the neck of the InternImage-S + BiFPN model, with corresponding object detection predictions for image `ID 171382`.



(a) `Ablation CAM` applied to Layer 1.

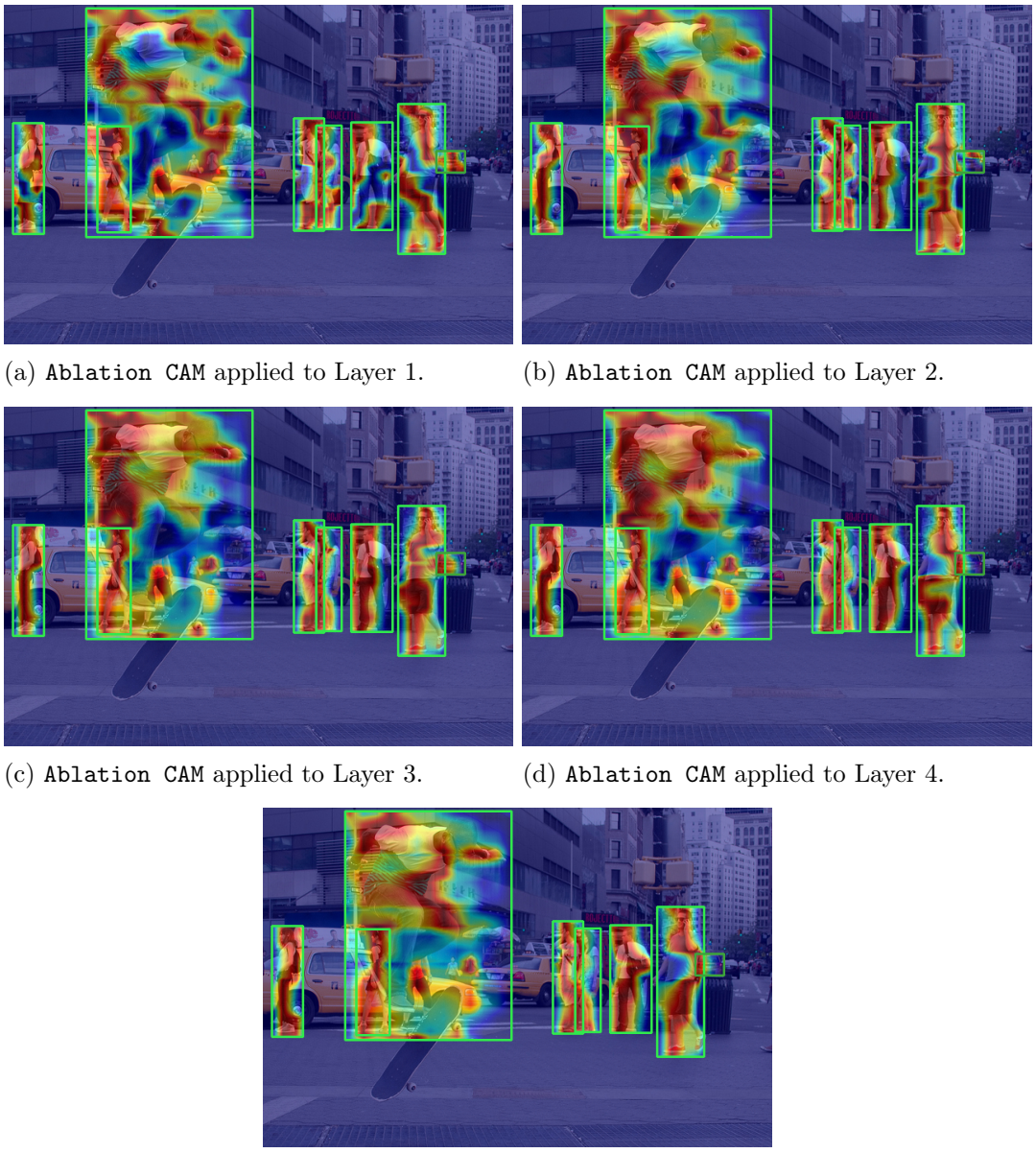
(b) `Ablation CAM` applied to Layer 2.

(c) `Ablation CAM` applied to Layer 3.

(d) `Ablation CAM` applied to Layer 4.

(e) `Ablation CAM` applied to Layer 5.

Figure 5.4: `Ablation CAM` applied to the five layers in the neck of the InternImage-S + LGFFDM model, with corresponding object detection predictions for image `ID 171382`.



(a) `Ablation CAM` applied to Layer 1.

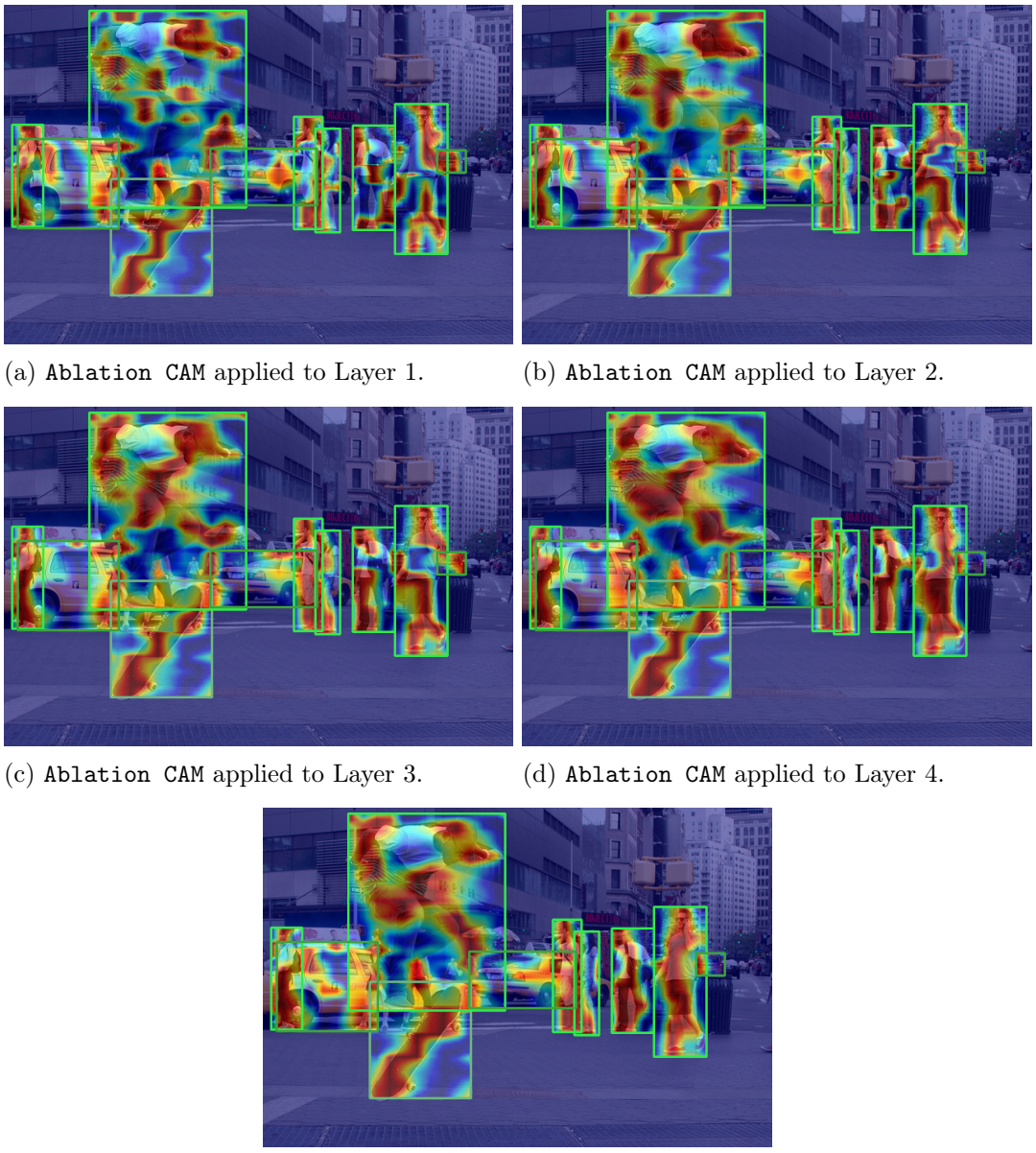
(b) `Ablation CAM` applied to Layer 2.

(c) `Ablation CAM` applied to Layer 3.

(d) `Ablation CAM` applied to Layer 4.

(e) `Ablation CAM` applied to Layer 5.

Figure 5.5: `Ablation CAM` applied to the five layers in the neck of the ConvNeXt-S + BiFPN model, with corresponding object detection predictions for image `ID 171382`.



(a) `Ablation CAM` applied to Layer 1.

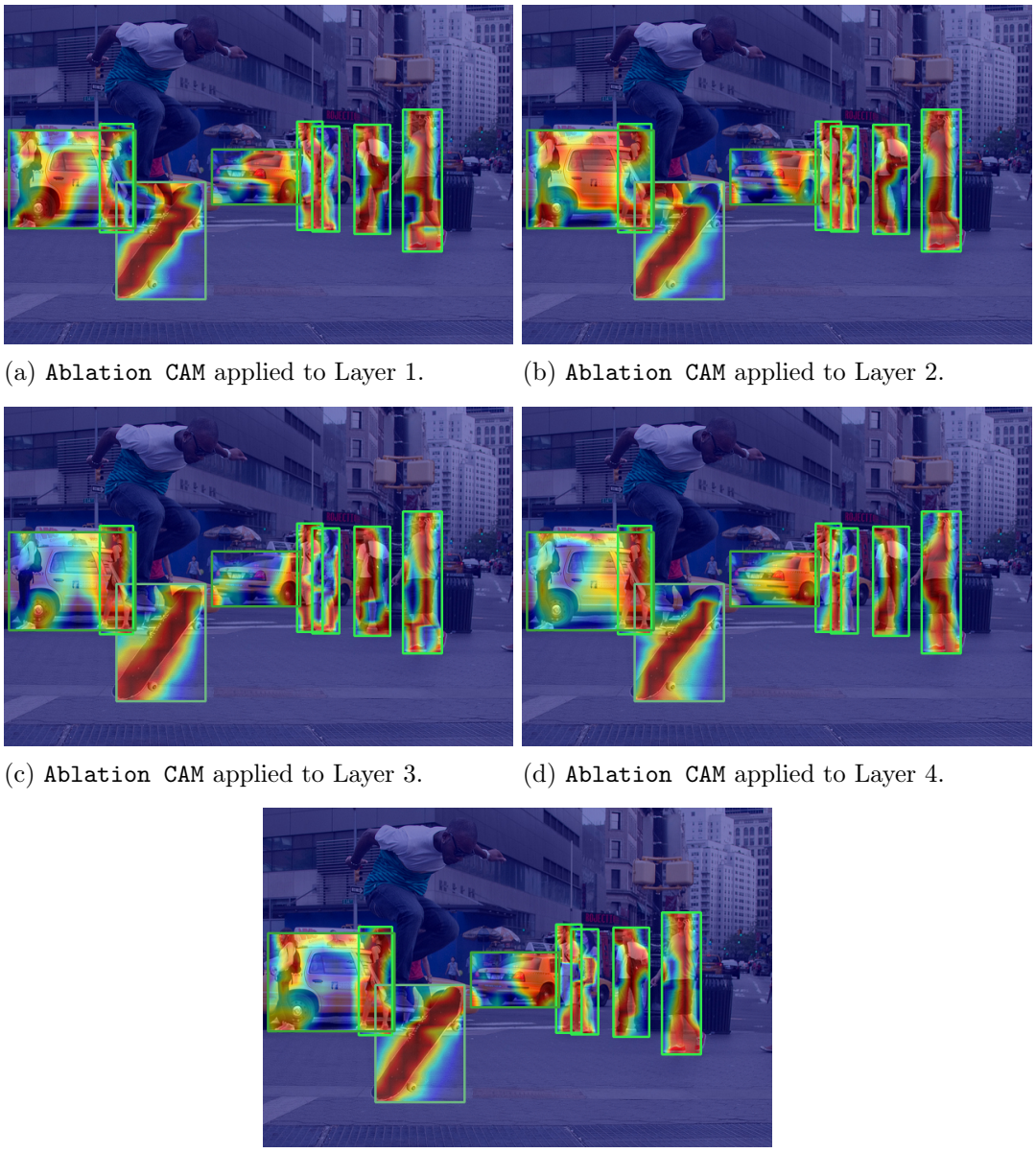
(b) `Ablation CAM` applied to Layer 2.

(c) `Ablation CAM` applied to Layer 3.

(d) `Ablation CAM` applied to Layer 4.

(e) `Ablation CAM` applied to Layer 5.

Figure 5.6: `Ablation CAM` applied to the five layers in the neck of the ConvNeXt-S + LGFFDM model, with corresponding object detection predictions for image `ID 171382`.



(a) `Ablation CAM` applied to Layer 1.

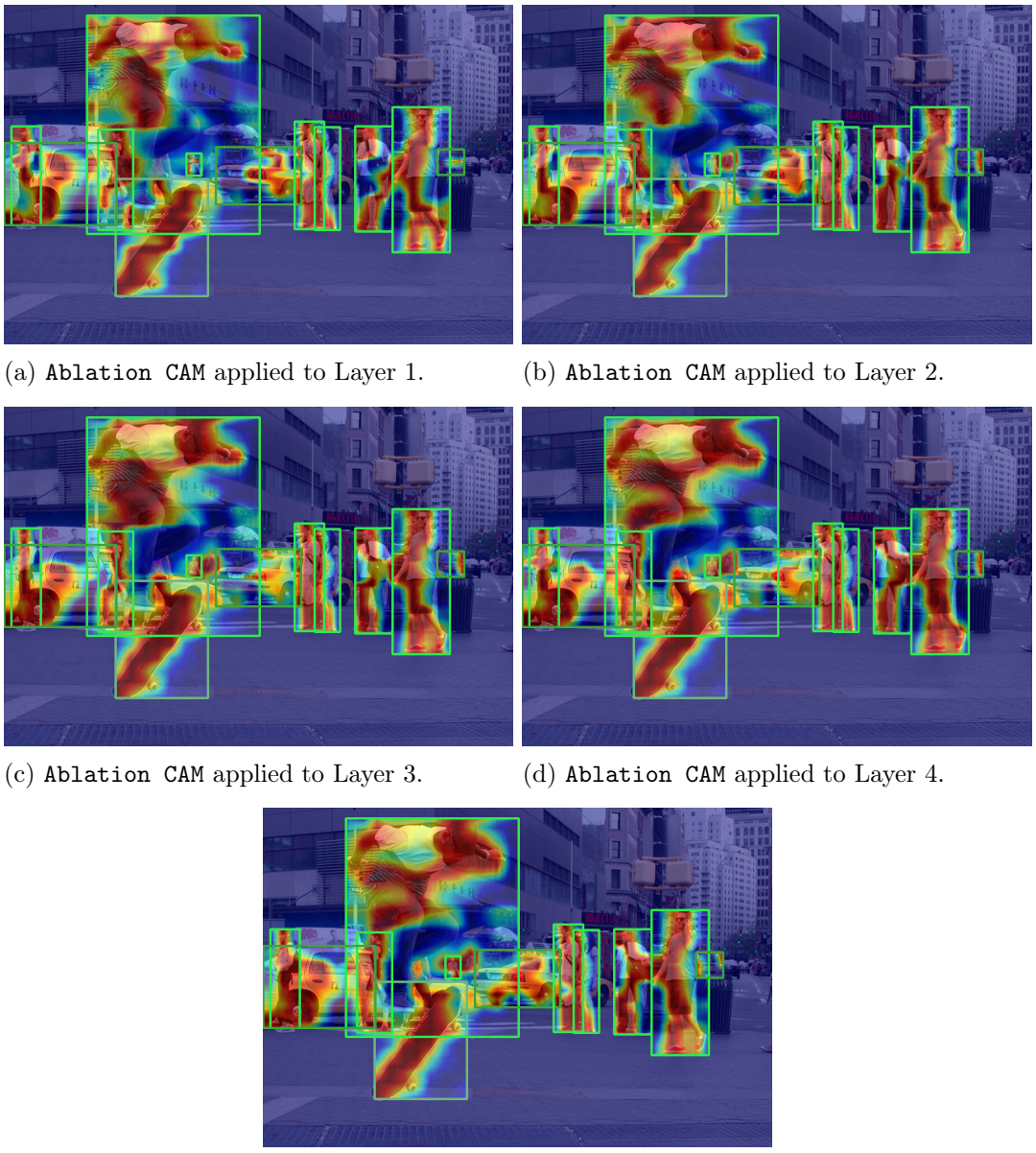
(b) `Ablation CAM` applied to Layer 2.

(c) `Ablation CAM` applied to Layer 3.

(d) `Ablation CAM` applied to Layer 4.

(e) `Ablation CAM` applied to Layer 5.

Figure 5.7: `Ablation CAM` applied to the five layers in the neck of the EfficientNetV2-M + BiFPN model, with corresponding object detection predictions for image `ID 171382`.



(a) `Ablation CAM` applied to Layer 1.

(b) `Ablation CAM` applied to Layer 2.

(c) `Ablation CAM` applied to Layer 3.

(d) `Ablation CAM` applied to Layer 4.

(e) `Ablation CAM` applied to Layer 5.

Figure 5.8: `Ablation CAM` applied to the five layers in the neck of the EfficientNetV2-M + LGFFDM model, with corresponding object detection predictions for image `ID 171382`.

5.3 LGFFEM Results

5.3.1 Metric Evaluation Results

As mentioned in Section 5.1.3, we present the metric results for accuracy retrieval across two scopes:

- Scope 1: Accuracy retrieval over the test datasets $\mathcal{R}$Oxford and $\mathcal{R}$Paris [51] using the five strategies A, B.1, B.2, B.3, and C.
- Scope 2: Accuracy retrieval over the test dataset Kimia Patch24C [55], using strategies A, C, and the best model resulting from strategies B.1, B.2, B.3.

Metrics Results Scope 1

As standard practice for image retrieval tasks, we present the mean average precision (mAP) for performance evaluation using the state-of-the-art test datasets $\mathcal{R}$Oxford and $\mathcal{R}$Paris. The mAP is presented under different difficulty levels, defined by treating labels (easy, hard, unclear) as positive or negative, or ignoring them. The results are shown using the following evaluations:

- Easy (E): Easy images are treated as positive, while hard and unclear images are ignored.
- Medium (M): Easy and hard images are treated as positive, while unclear images are ignored.
- Hard (H): Hard images are treated as positive, while easy and unclear images are ignored.

The evaluation accuracy results using the five training strategies A, B.1, B.2, B.3, and C are presented in Table 5.4.

Metrics Results Scope 2

To evaluate the domain specification approach using histopathology images, we present the accuracy results η_p, η_w, and η_{tot} for the three remaining training strategies: A, B.1 (the best of the B strategies), and C, as

Table 5.4: Retrieval accuracy for strategies A, B.1, B.2, B.3, and C on the $\mathcal{R}$Oxford and $\mathcal{R}$Paris datasets.

		Easy		**Medium**		**Hard**	
Strategy	**Pre-training Data**	$\mathcal{R}$Oxf	$\mathcal{R}$Par	$\mathcal{R}$Oxf	$\mathcal{R}$Par	$\mathcal{R}$Oxf	$\mathcal{R}$Par
A	LGFFEM + IN-1K	19.78	54.64	14.6	43.68	3.56	20.6
B.1	LGFFEM + IN-1K + PanNune (m=28.6)	10.62	45.27	9.70	36.72	2.97	14.96
B.2	LGFFEM + IN-1K + PanNune (m=17.2)	7.86	28.69	7.82	24.86	2.30	10.16
B.3	LGFFEM + IN-1K + PanNune (m=5.73)	10.74	31.31	9.95	24.61	2.04	7.98
C	LGFFEM + IN-1K + PanNune + Kimia	2.70	7.63	3.39	8.67	1.18	4.48

shown in Table 5.5. These results were obtained using the test dataset of Kimia Patch24C.

Table 5.5: Retrieve accuracy(%) for the strategies A, B.1 and C.

Strategy	**Pre-training Data**	η_p	η_w	η_{tot}
A	IN-1K	72.08	74.37	53.6
B.1	IN-1K + PanNuke	77.36	79.28	61.33
C	IN-1K + PanNuke + Kimia	**99.40**	**99.47**	**98.87**

For the baseline comparison, we evaluated our proposed model against the models used in [55] and a more recent work [74]. The accuracy results of these two baseline models, along with our proposed model, are shown in Table 5.6.

Table 5.6: Retrieve accuracy(%) for baselines models and the better model obtained from the strategy C.

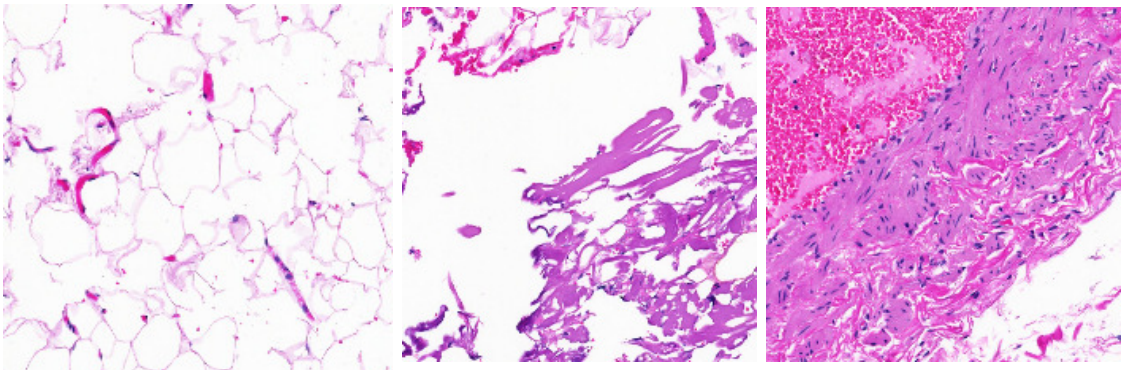
Method	η_p	η_w	η_{tot}
DenseNet 121 [55]	95.92	95.51	91.62
MA + MS-loss [74]	97.89	97.00	94.95
LGFFEM [ours]	**99.40**	**99.47**	**98.87**

5.3.2 Ablation CAM Results

Similar to the LGFFDM results, we will examine the Class Activation Mapping (CAM) across three layers of the neck, utilizing the Ablation CAM visual explanation technique. In this case, the results aim to visualize the behavior of the feature fusion operations in the outer aggregation fusion nodes P_{1_2}, P_{2_2}, P_{3_2}, and P_{4_2} as described in Equation (4.2.2).

We selected the first image from the class set *S0* of the test dataset of Kimia Patch24C as a query image and retrieved the first two results using the FAISS library. These three images, one query image and two retrieved images, are depicted in Figure 5.9.

For the three selected images, we applied Ablation CAM to the final model used in Strategy C, across the three layers of the neck and the four outer aggregation fusion nodes independently, as well as a complete collapse of the four nodes projection. The results of these visualizations are systematically represented in Figures 5.10 to 5.12 for the first retrieved image (ID `S0-2`), and Figures 5.3 to 5.8 for the second retrieved image (ID `S0-5`), corresponding to the three layers of the neck model.

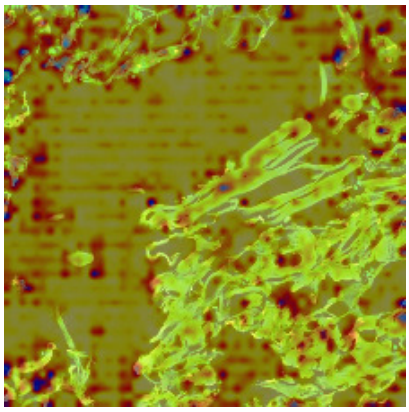


(a) Query image ID `S0-1`.

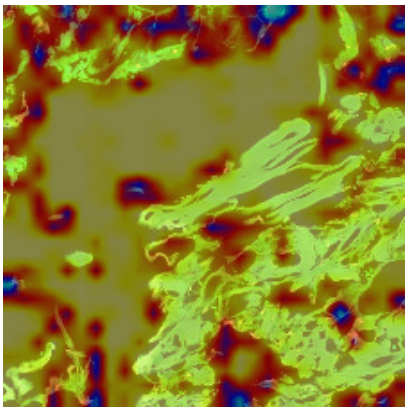
(b) First image retrieved ID `S0-2`.

(c) Second image retrieved ID `S0-5`.

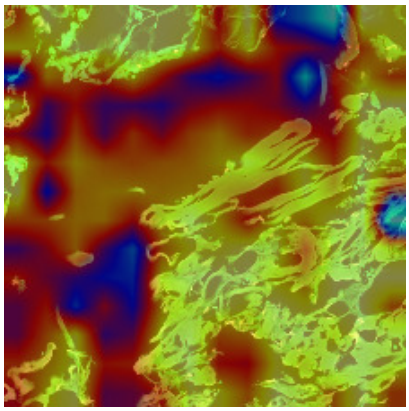
Figure 5.9: Query image selected for the class set *S0* and their firsts two retrieve images from the Kimia Patch24C dataset.



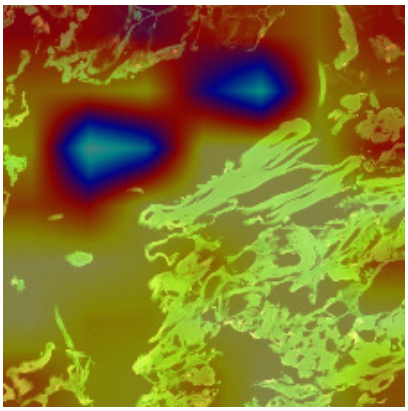
(a) `Ablation CAM` applied to the outer aggregation fusion node P_{1_2} in Layer 1.



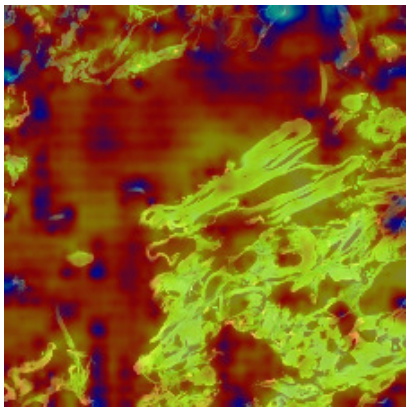
(b) `Ablation CAM` applied to the outer aggregation fusion node P_{2_2} in Layer 1.



(c) `Ablation CAM` applied to the outer aggregation fusion node P_{3_2} in Layer 1.

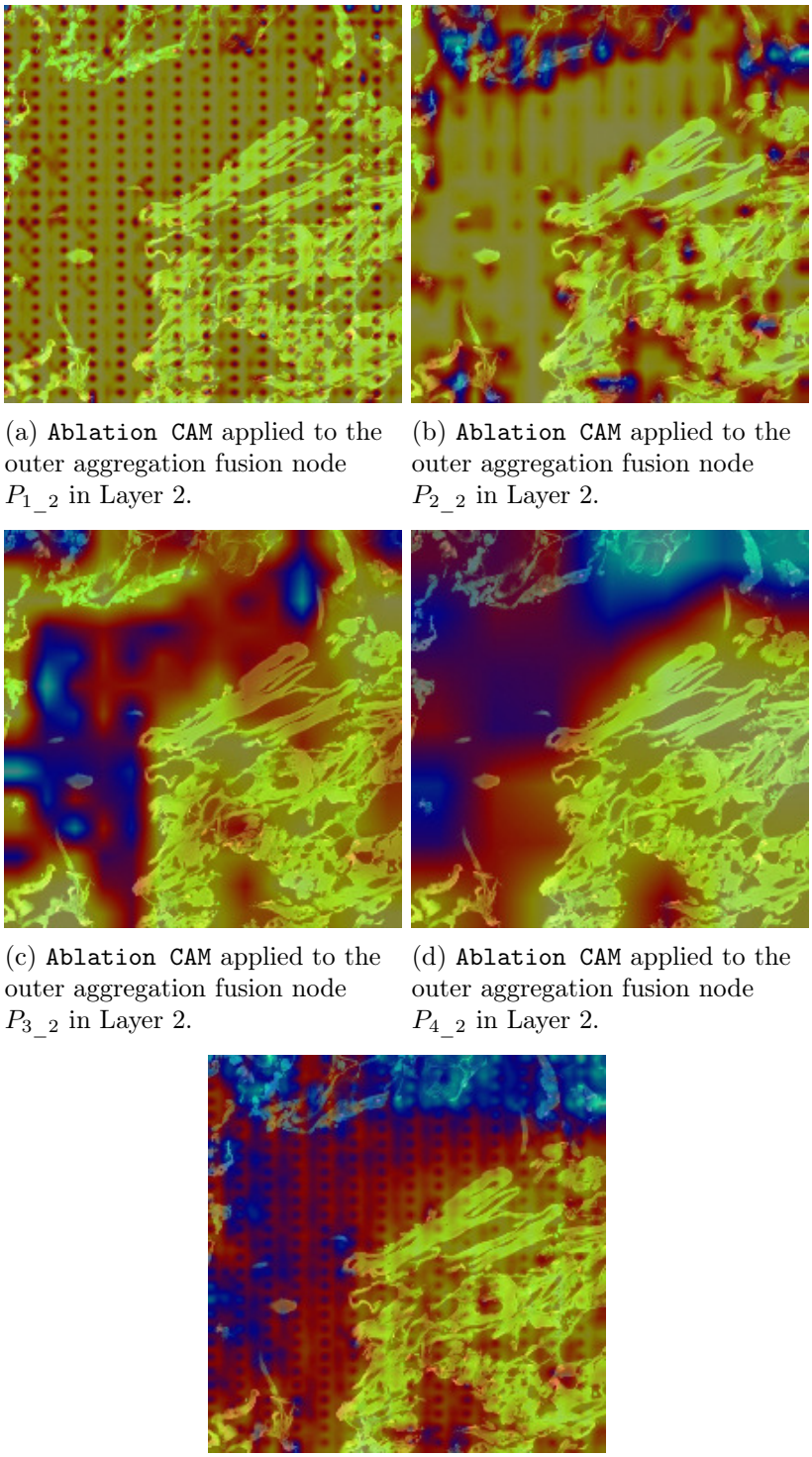


(d) `Ablation CAM` applied to the outer aggregation fusion node P_{4_2} in Layer 1.



(e) `Ablation CAM` applied to the collapse of all outer aggregation fusion nodes in Layer 1.

Figure 5.10: `Ablation CAM` applied to first layer of the neck used in the strategy C for the first image retrieved ID `S0-2`.



(a) `Ablation CAM` applied to the outer aggregation fusion node P_{1_2} in Layer 2.

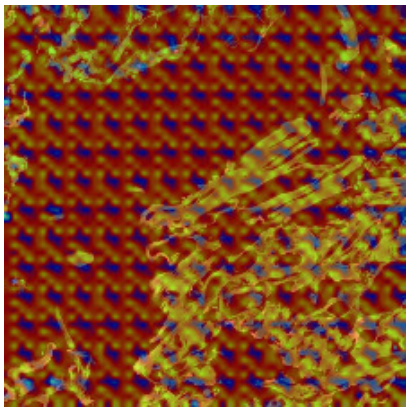
(b) `Ablation CAM` applied to the outer aggregation fusion node P_{2_2} in Layer 2.

(c) `Ablation CAM` applied to the outer aggregation fusion node P_{3_2} in Layer 2.

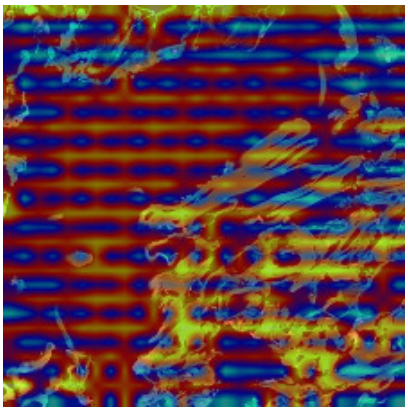
(d) `Ablation CAM` applied to the outer aggregation fusion node P_{4_2} in Layer 2.

(e) `Ablation CAM` applied to the collapse of all outer aggregation fusion nodes in Layer 2.

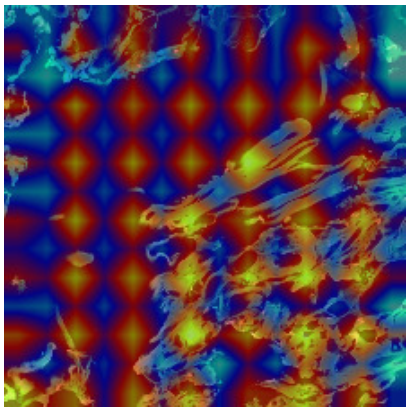
Figure 5.11: `Ablation CAM` applied to second layer of the neck used in the strategy C for the first image retrieved ID `S0-2`.



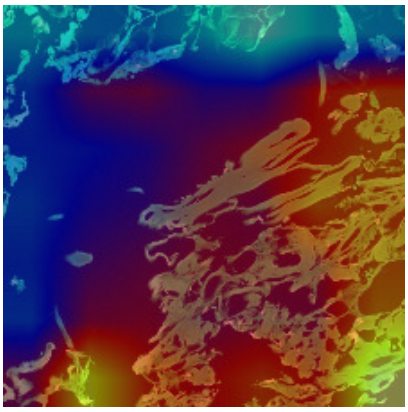
(a) `Ablation CAM` applied to the outer aggregation fusion node P_{1_2} in Layer 3.



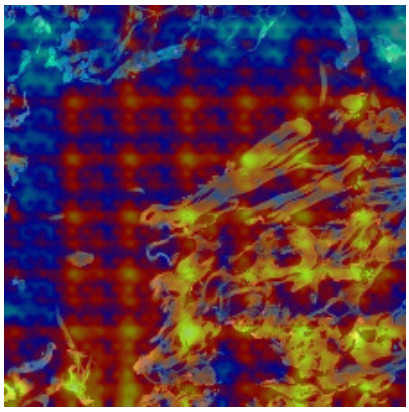
(b) `Ablation CAM` applied to the outer aggregation fusion node P_{2_2} in Layer 3.



(c) `Ablation CAM` applied to the outer aggregation fusion node P_{3_2} in Layer 3.

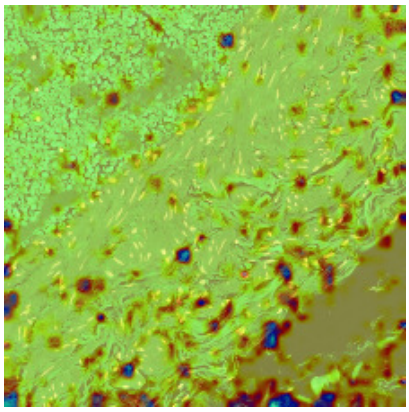


(d) `Ablation CAM` applied to the outer aggregation fusion node P_{4_2} in Layer 3.

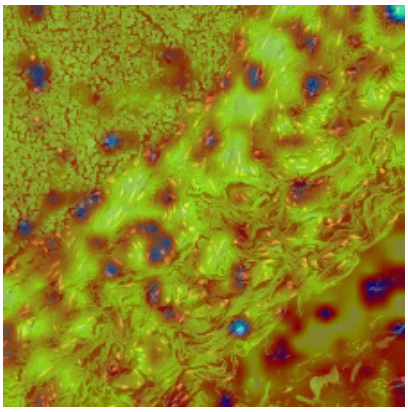


(e) `Ablation CAM` applied to the collapse of all outer aggregation fusion nodes in Layer 3.

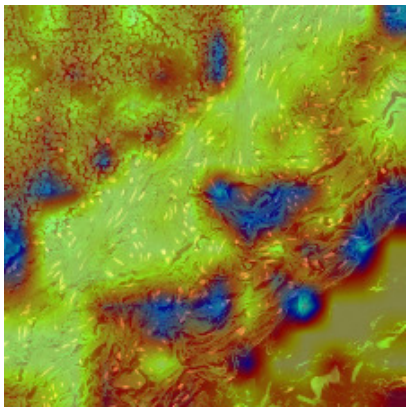
Figure 5.12: `Ablation CAM` applied to third layer of the neck used in the strategy C for the first image retrieved ID `S0-2`.



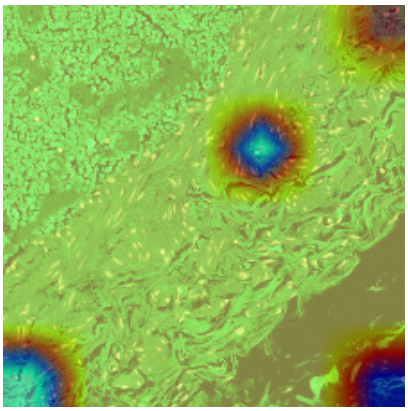
(a) `Ablation CAM` applied to the outer aggregation fusion node P_{1_2} in Layer 1.



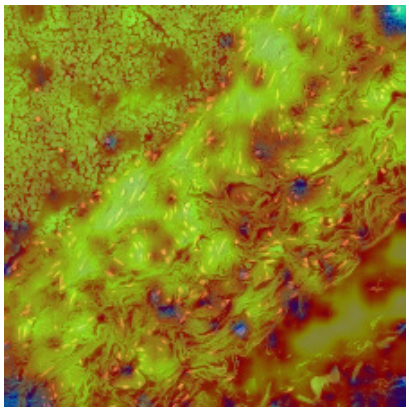
(b) `Ablation CAM` applied to the outer aggregation fusion node P_{2_2} in Layer 1.



(c) `Ablation CAM` applied to the outer aggregation fusion node P_{3_2} in Layer 1.

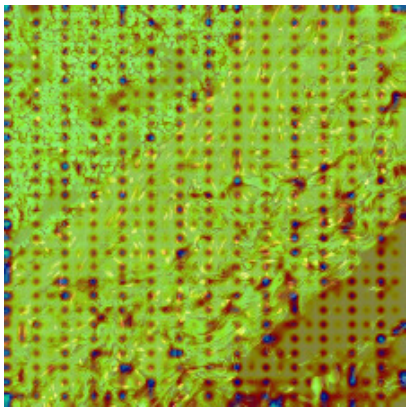


(d) `Ablation CAM` applied to the outer aggregation fusion node P_{4_2} in Layer 1.

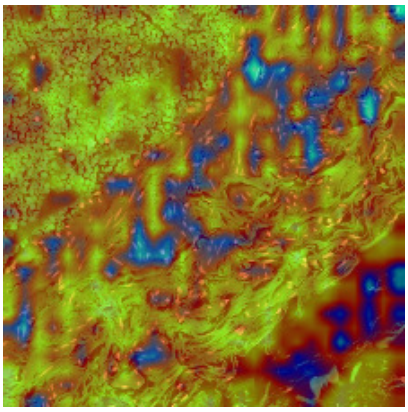


(e) `Ablation CAM` applied to the collapse of all outer aggregation fusion nodes in Layer 1.

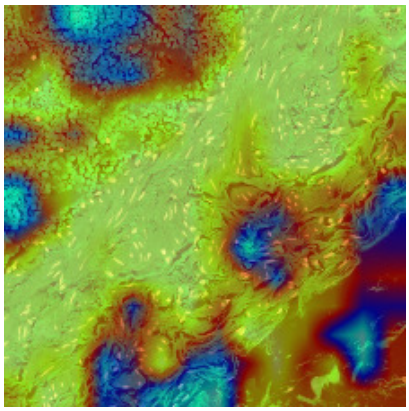
Figure 5.13: `Ablation CAM` applied to first layer of the neck used in the strategy C for the second image retrieved ID `S0-5`.



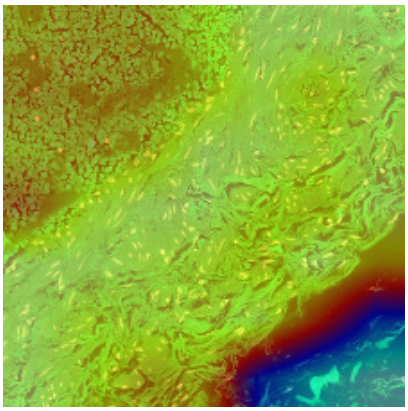
(a) `Ablation CAM` applied to the outer aggregation fusion node P_{1_2} in Layer 2.



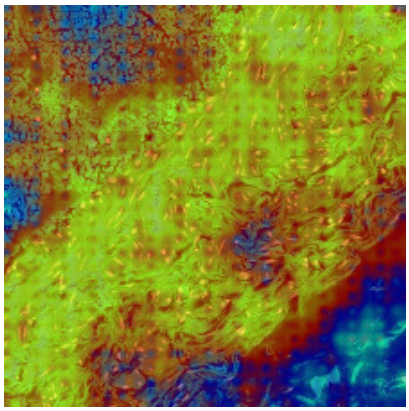
(b) `Ablation CAM` applied to the outer aggregation fusion node P_{2_2} in Layer 2.



(c) `Ablation CAM` applied to the outer aggregation fusion node P_{3_2} in Layer 2.

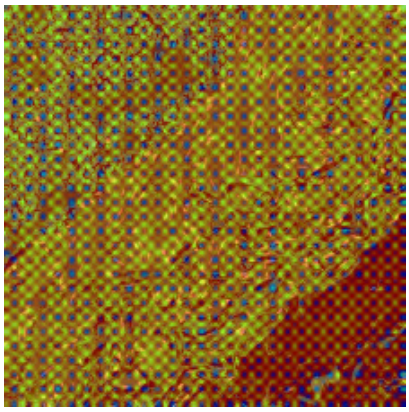


(d) `Ablation CAM` applied to the outer aggregation fusion node P_{4_2} in Layer 2.

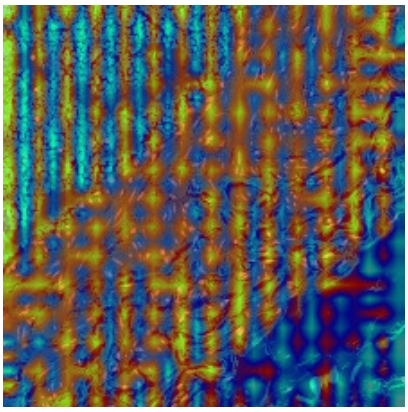


(e) `Ablation CAM` applied to the collapse of all outer aggregation fusion nodes in Layer 2.

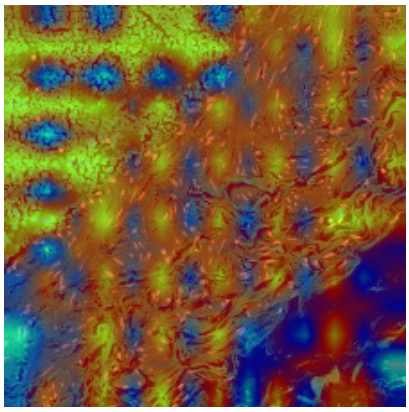
Figure 5.14: `Ablation CAM` applied to second layer of the neck used in the strategy C for the second image retrieved ID `S0-5`.



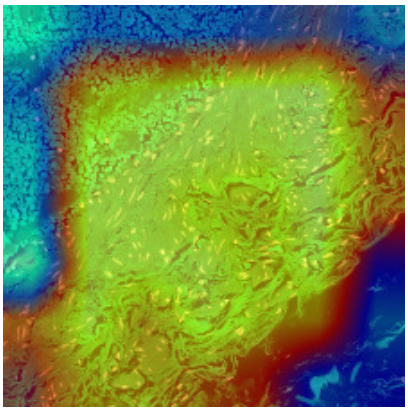
(a) `Ablation CAM` applied to the outer aggregation fusion node P_{1_2} in Layer 3.



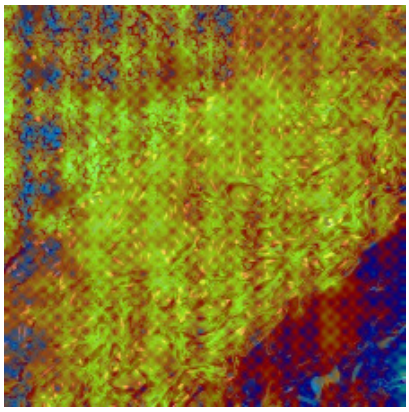
(b) `Ablation CAM` applied to the outer aggregation fusion node P_{2_2} in Layer 3.



(c) `Ablation CAM` applied to the outer aggregation fusion node P_{3_2} in Layer 3.



(d) `Ablation CAM` applied to the outer aggregation fusion node P_{4_2} in Layer 3.

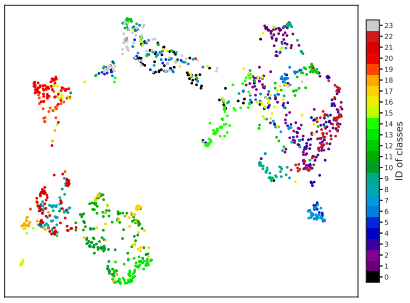


(e) `Ablation CAM` applied to the collapse of all outer aggregation fusion nodes in Layer 3.

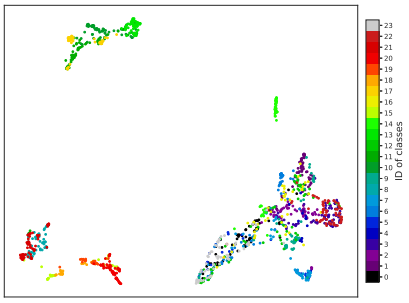
Figure 5.15: `Ablation CAM` applied to third layer of the neck used in the strategy C for the second image retrieved ID `S0-5`.

5.3.3 Embeddings 2D Projections

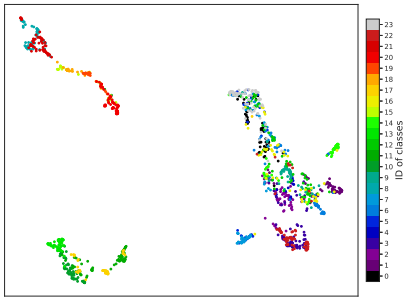
To visualize the discriminative clustering of different class types in a 2D space, we developed distinct visualization approaches for the class clusters in each of the five training strategies. For each strategy, we obtained the vector image descriptor f_e for each image in the Kimia Patch24C test dataset. We then applied the UMAP technique [47] to project the high-dimensional space of f_e to 2D. The resulting 2D plots of the image descriptors for each of the five strategies are shown in Figure 5.16.



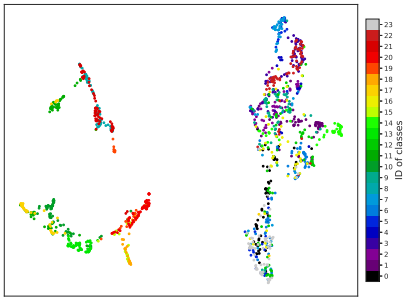
(a) Visualization of the 2D projection of embeddings from the Kimia Patch24C dataset using strategy A.



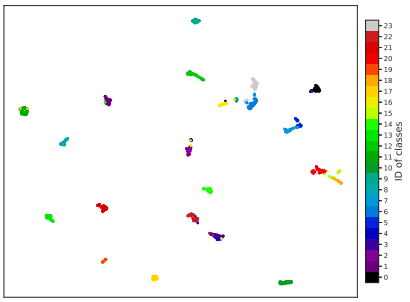
(b) Visualization of the 2D projection of embeddings from the Kimia Patch24C dataset using strategy B.1.



(c) Visualization of the 2D projection of embeddings from the Kimia Patch24C dataset using strategy B.2.



(d) Visualization of the 2D projection of embeddings from the Kimia Patch24C dataset using strategy B.3.



(e) Visualization of the 2D projection of embeddings from the Kimia Patch24C dataset using strategy C.

Figure 5.16: Visualization of the 2D projection of embeddings from the Kimia Patch24C dataset for the five strategies. Each dot represents an image, and each color represents a class.

Chapter 6

Discussion and Analysis

6.1 LGFFDM Analysis

6.1.1 Metric Evaluation Analysis

Upon scrutiny of the aggregated data from the Table 5.3, it becomes evident that networks employing the proposed LGFFDM neck architecture demonstrated superior performance in both bounding box and mask predictions, when compared to those networks utilizing the conventional BiFPN neck. More specifically, the architecture that most proficiently predicted bounding boxes was a combination of EfficientNetV2-M [63] with LGFFDM. Conversely, the least effective architecture for this particular task was identified as InternImage-S [72] combined with BiFPN [65]. In terms of mask prediction, the ConvNeXt-S network [46] paired with LGFFDM produced the most favorable outcomes, while the least effective results were similarly attributed to the InternImage-S network [72] with neck BiFPN [65].

It is noteworthy to address the implications of architectural choices on model complexity. The integration of a greater number of hyperparameters in the proposed LGFFDM neck substantially elevates the model's depth-related complexity as compared to its BiFPN counterpart. This suggests that a more expansive neck architecture, such as LGFFDM, manifests a higher degree of computational sophistication than a less expansive structure like BiFPN. However, this elevated complexity is accompanied by more intricate internal representations within the neck architecture.

Consequently, this demands additional training time and computational resources, aligning with the observations put forth in previous work [6].

6.1.2 Ablation CAM visual explanation Image `171382`

In the present study, Image ID `171382` serves as a particularly challenging example for evaluating the efficacy of instance detection across six pre-trained models. This complexity arises from a confluence of factors such as the sheer number of instances, variability in their dimensions and geometries, or the degree of overlapping among them. Figures 5.3 to 5.8 provide visual representations to support our observations. Notably, it is exclusively the models incorporating the LGFFDM neck that demonstrate a superior ability in detecting a diverse array of instances, beyond the over-represented `'person'` class.

In the specific of Figures 5.3 and 5.4, models employing both InternImage-S backbone and BiFPN and LGFFDM necks primarily identify instances of the `'person'` class. Yet, the model featuring the LGFFDM neck exhibits the capability to discern `'person'` instances across a range of scales. In contrast, the model with the BiFPN neck only effectively identifies `'person'` instances when they are confined to a similar area and pose. This suggests that the BiFPN-based model is more prone to neglecting underrepresented contextual features within the image.

Further scrutiny of Figures 5.5 and 5.6 reveals nuanced behavior from the ConvNeXt-S + BiFPN and ConvNeXt-S + LGFFDM models. Again, the model with the LGFFDM neck outperforms its BiFPN counterpart in terms of instance detection. Interestingly, the LGFFDM model initiates with dispersed focal regions in its initial layers, particularly noticeable in the `'person'` instance depicted performing a jumping gesture over a `'skateboard'` instance, unlike other `'person'` instances where the gestures are more homogenous.

Lastly, an examination of Figures 5.7 and 5.8, which correspond to the EfficientNetV2-M + BiFPN and EfficientNetV2-M + LGFFDM models,

respectively, corroborates the aforementioned pattern. Here again, the LGFFDM model identifies a greater number of instances. However, it is only in the model's advanced layers that the fuse of high-level features becomes more manifest, thereby accentuating key discriminative regions such as the limbs of the `'person'` instances.

6.2 LGFFEM Analysis

Due to the best results from LGFFDM coming from the model combination ConvNeXt-S + LGFFDM as show in the previous section, all subsequent experiments for the LGFFEM model were conducted using this model.

6.2.1 Metric Evaluation Analysis

The main objective of applying the Scope 1 with five different models, each trained with three different transfer learning techniques, is to understand how domain generalization decrease as the domain-specific dataset expands. In this context, the expected behavior is a decrease in accuracy over the test datasets $\mathcal{R}$Oxford and $\mathcal{R}$Paris. This decline occurs when the general domain knowledge provided by the ImageNet-1k dataset is lost, as the models are trained using domain-specific datasets like PanNuke or Kimia Patch24C. This expected behavior of decrease in the accuracy values is confirm successfully in result presents in the Table 5.4.

The Scope 2 must show the inverse behavior of the Scope 1, that's means the accuracy values increase for a specific histological domain when the domain goes deeper in the images types. As seen in the Table 5.5, the accuracy metrics improve when the proposed model is trained on different types of image domains. In detail, the domain of classic images from ImageNet-1k shows a lower accuracy compared to training using two domain-specific datasets, PanNuke and Kimia Patch24C. Furthermore, high accuracy is achieved when transferring learning from the first two datasets, ImageNet-1k and PanNuke, in the training process of strategy C for the domain-specific Kimia Patch24C.

For the baseline analysis, we compared our proposed model with the model used in the work [55] and a more recent work [74] model. The accuracy results of these two baseline models and our proposed model are shown in Table 5.6. Our model, trained with strategy C, surpassed the accuracy achieved by the two baseline models.

6.2.2 The influence of the angular margin hyperparameter

A parallel objective that must be addressed is the inclusion of the m hyperparameter in the five strategies. In this case, m corresponds to the angular margin in the feature space, as described in Sub-center ArcFace [13] and CosFace [71] original work. As m increases, the angular margin between different classes is amplified.

As explained in Section 5.1.3, the five strategies A, B.1, B.2, B.3, and C utilise the m hyperparameter. For strategies A and C, the value was set to $m = 28.6$ degrees. However, strategies B use different values: $m = 28.6$ (B.1), $m = 17.2$ (B.2), and $m = 5.73$ (B.3) degrees, to explore how the angular margin affects the results. For the methods with the dataset ImageNet-1k + PanNuke in the B strategies, the results can be found in the second combined row of Table 5.4.

From these results, it is interesting to note that the best values are achieved with the margin set to $m = 28.6$ degrees, while the worst values are seen with margins less than $m = 28.6$ degrees ($m = 17.2$ and $m = 5.73$); similar to the findings in [71], as m increases, the accuracy values improve.

6.2.3 Explanation with Ablation CAM

Based on preliminary clinical analysis, the query image represents mature adipose tissue stained with Hematoxylin and Eosin (H&E) and Masson's trichrome. In the image with ID `S0-2`, mature adipose tissue is visible in the upper left corner, while fibroblastic proliferation is observed in the

bottom right corner. In the image with ID `S0-5`, most of the image is composed of proliferating fibroblasts and type II pneumocytes, with a mature adipose area observed in the bottom right corner.

On the other hand, to explore the tissue morphology and cell structure of the image, we applied Ablation CAM visual explanation over the outer aggregation fusion nodes P_{1_2}, P_{2_2}, P_{3_2}, and P_{4_2} from Equation (4.2.2). The objective was to understand how the model interprets these characteristics.

Ablation CAM visual explanation image retrieved ID `S0-2`

In the first layer, nodes P_{1_2} and P_{2_2} do not identify any morphology or structural types in the image. Despite this, nodes P_{3_2} and P_{4_2} recognize some parts of the tissue, as indicated by the red gradient areas, while avoiding the immunohistochemical zones, shown as blue gradient areas. However, they still do not identify any specific morphology or structure. The combined results of the four nodes confirm this behavior, indicating that the initial layer is not effective in capturing the relevant tissue characteristics.

In the second layer, the behavior of nodes P_{1_2} and P_{2_2} remains consistent, as they again fail to identify the morphology or structure type in the image. These nodes only display the initialization of the fusion feature in random areas, such as the texture points in node P_{1_2}. This suggests that these nodes are not yet tuned to detect meaningful features in the tissue. However, node P_{3_2} shows a significant improvement, identifying mature adipose tissue in the upper left corner and the morphology of fibroblastic areas in the bottom right corner. Additionally, this node effectively avoids the immunohistochemical zones, indicating a more refined feature detection capability. Node P_{4_2}, on the other hand, loses its focus on the mature adipose tissue but maintains its attention on the fibroblast area. The combined results show that node P_{4_2} has a greater influence than the other three nodes in this layer, suggesting that it may play a dominant role in feature detection at this stage.

In the third layer, nodes P_{1_2}, P_{2_2}, and P_{3_2} exhibit delayed and sluggish fusion behavior, similar to what is described in Section 6.1.2. This behavior indicates that these nodes are not effectively integrating the detected features into a coherent representation. Only node P_{4_2} focuses on the morphology zone, particularly the fibroblast area, demonstrating a more targeted and effective feature detection. The combined results from this layer display the delayed and sluggish fusion behavior across all nodes, emphasizing the challenges in achieving effective feature integration. This suggests that further refinement and tuning of the model are needed to improve its ability to accurately identify and integrate tissue characteristics across multiple layers.

Ablation CAM visual explanation image retrieved ID `S0-5`

The image with ID `S0-5` is expected to be more complex for the model to analyze because most of the morphology areas are composed of fibroblasts, type II pneumocytes, and a small portion of mature adipose tissue. This complexity is evident in the undefined fusion across the four nodes P_{1_2}, P_{2_2}, P_{3_2}, and P_{4_2} in layer 1, where it is impossible to identify any specific tissue morphology or cell structure.

However, this behavior changes from layer 2 onward. In this layer, nodes P_{1_2} and P_{2_2} display delayed and sluggish fusion detection areas, but node P_{3_2} successfully detects fibroblasts, type II pneumocytes, and mature adipose areas. Node P_{4_2} also successfully detects fibroblasts and type II pneumocytes, but, similar to its behavior in layer 2 for image ID `S0-2`, it loses focus on the mature adipose tissue area. In this layer, nodes P_{3_2} and P_{4_2} are more relevant than nodes P_{1_2} and P_{2_2}, as shown by the combined results of all nodes.

In the third layer, nodes P_{1_2}, P_{2_2}, and P_{3_2} again exhibit delayed and sluggish fusion behavior. Node P_{4_2} focuses its attention on the type II pneumocytes areas, avoiding the other areas. When the nodes are combined, the focused areas are the fibroblast and type II pneumocyte zones.

6.2.4 Visualization of Learned Embeddings

To support the above results, a projection in a 2D space of the embeddings retrieved by our model across the five strategies is presented in Figure 5.16. This figure shows the clusters associated with the 24 different classes of tissues in the test dataset.

It is possible to observe how domain generalization diminishes as the training transfer method progresses. In the projection plot of strategy A, only a few clusters for the classes are visible, indicating limited separation and the predominance generalization. Meanwhile, the visualization for strategy B shows more defined clusters, suggesting improved differentiation between tissue classes. Finally, the visualization for strategy C demonstrates perfect cluster formation for each of the 24 different classes of tissues in the test dataset, confirming the effectiveness of the domain-specific training. This progression underscores the importance of tailored training strategies in enhancing model performance and achieving precise classification across diverse tissue types.

Chapter 7

Conclusion

This research undertakes a comprehensive analysis of feature fusion behaviors, focusing on both local and global feature fusion variables. Central to this investigation is the development and implementation of the models **Local-Global Feature Fusion Detection Model (LGFFDM)** and **Local-Global Feature Fusion Embedding Model (LGFFEM)**, using the proposed *Local-Global Feature Fusion Neck (LGFFN)*. Through a rigorous comparison with the conventional *Bi-Feature Pyramid Network* (BiFPN), the LGFFN demonstrates markedly superior performance in various tasks, such as object detection and extracting image descriptor embeddings.

The LGFFDM was proposed to validate the strategy of local-global feature fusion, while the principal model, LGFFEM, was designed to address the hypothesis of this investigation.

To achieve this initial validation using LGFFDM, our empirical findings reveal that models incorporating the LGFFDM architecture surpass those utilizing the standard BiFPN architecture, particularly in the realm of bounding box and mask predictions.

Based on our initial model and its results, we designed and implemented a unified model framework for extracting image descriptor embeddings from histopathological images using multi-scale local-global fused features. The results achieved by LGFFEM indicate that the proposed method can sur-

pass baseline models with a new training strategy and multi-scale local-global fused features. This proposed method is designed with fewer parameters compared to classical pre-trained models, allowing it to be trained on more domain-specific medical images without the necessity of using complex computer resources.

In particular, LGFFEM shows promising results in addressing important study questions, such as: how domain-specific training affects the accuracy of the model, how tissue morphology and cell structure can be interpreted by local-global feature fusion, and how the generation of visual descriptor embeddings can preserve high-order image semantic structure.

In summary, both the general and specific objectives of this research were fully realized. Quantitative findings were further supported through the utilization of Ablation CAM and accuracy metrics, providing a nuanced understanding of variations in model performance. Finally, we confirm our hypothesis: multiple feature vectors can be generated to enhance the robustness and efficiency of CMBIR systems for histopathological images, leveraging a multi-scale local-global feature fusion approach.

7.1 Future Work

Several avenues for future research are delineated below:

- As discussed in Section 5.1.3, we opted to resize images from the COCO dataset to the original dimensions of 224×224 as used in the ImageNet dataset. A more comprehensive analysis should explore varying input sizes to understand the behavior of multi-scale variations in the images. Alternatively, one may choose to not resize the dataset, using the original COCO image dimensions.
- Further studies in domain-specific areas, such as X-ray and pap smear images, are warranted to explore the capability of the LGFFEM. This exploration aims to develop a comprehensive web-based CBMIR system for various medical imaging specialists.

- Based on the discussion in Sections 6.2.3 and 6.2.3, it is possible to explore a new vector image descriptor f_e by combining only select outer aggregation fusion nodes, such as P_{3_2} or P_{4_2}. This approach avoids the more premature fusion nodes like P_{1_2} and P_{2_2}, resulting in a less parameter-complex Embedding Head, as discussed in Section 4.2.3.

Bibliography

[1] Abdelsamea, M. M., Zidan, U., Senousy, Z., Gaber, M. M., Rakha, E., and Ilyas, M. A survey on artificial intelligence in histopathology image analysis. *WIREs Data Mining and Knowledge Discovery 12*, 6 (2022), e1474.

[2] Adelson, E. H., Anderson, C. H., Bergen, J. R., Burt, P. J., and Ogden, J. M. 1984, Pyramid methods in image processing. *RCA Engineer 29*, 6 (1984), 33–41.

[3] Ando, D. M., McLean, C. Y., and Berndl, M. Improving phenotypic measurements in high-content imaging screens. *bioRxiv* (2017).

[4] Arndt, J. *Matters Computational*. Springer Berlin Heidelberg, 2011.

[5] Babaie, M., Kalra, S., Sriram, A., Mitcheltree, C., Zhu, S., Khatami, A., Rahnamayan, S., and Tizhoosh, H. R. Classification and retrieval of digital pathology scans: A new dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (2017).

[6] Bengio, Y. Learning deep architectures for AI. *Foundations and Trends® in Machine Learning 2*, 1 (2009), 1–127.

[7] Buslaev, A., Iglovikov, V. I., Khvedchenya, E., Parinov, A., Druzhinin, M., and Kalinin, A. A. Albumentations: Fast and flexible image augmentations. *Information 11*, 2 (2020).

[8] Cao, B., Araujo, A., and Sim, J. Unifying deep local and global features for image search. In *Computer Vision – ECCV 2020* (2020).

[9] Chattopadhyay, A., Sarkar, A., Howlader, P., and Balasubramanian, V. N. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. *CoRR abs/1710.11063* (2017).

[10] Dai, Y., Gieseke, F., Oehmcke, S., Wu, Y., and Barnard, K. Attentional feature fusion. *CoRR abs/2009.14082* (2020).

[11] Dalal, N., and Triggs, B. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)* (2005), vol. 1, pp. 886–893 vol. 1.

[12] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition* (2009), pp. 248–255.

[13] Deng, J., Guo, J., Liu, T., Gong, M., and Zafeiriou, S. Sub-center arcface: Boosting face recognition by large-scale noisy web faces. In *Computer Vision – ECCV 2020* (2020).

[14] Desai, S., and Ramaswamy, H. G. Ablation-cam: Visual explanations for deep convolutional network via gradient-free localization. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)* (2020), pp. 972–980.

[15] Dumoulin, V., and Visin, F. A guide to convolution arithmetic for deep learning. *ArXiv e-prints* (mar 2016).

[16] Fu, R., Hu, Q., Dong, X., Guo, Y., Gao, Y., and Li, B. Axiom-based grad-cam: Towards accurate visualization and explanation of cnns. *CoRR abs/2008.02312* (2020).

[17] Fukushima, K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics 36*, 4 (Apr. 1980), 193–202.

[18] Gamper, J., Koohbanani, N. A., Benet, K., Khuram, A., and Rajpoot, N. Pannuke: an open pan-cancer histology dataset for nuclei instance segmentation and classification. In *European Congress on Digital Pathology* (2019), Springer, pp. 11–19.

[19] Gamper, J., Koohbanani, N. A., Graham, S., Jahanifar, M., Khurram, S. A., Azam, A., Hewitt, K., and Rajpoot, N. Pannuke dataset extension, insights and baselines. *arXiv:2003.10778* (2020).

[20] Ghiasi, G., Lin, T., Pang, R., and Le, Q. V. NAS-FPN: learning scalable feature pyramid architecture for object detection. *CoRR abs/1904.07392* (2019).

[21] Gildenblat, J., and contributors. Pytorch library for cam methods. `https://github.com/jacobgil/pytorch-grad-cam`, 2021.

[22] Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[23] Gu, W., Bai, S., and Kong, L. A review on 2d instance segmentation based on deep neural networks. *Image and Vision Computing 120* (2022), 104401.

[24] Gupta, S., and Tan, M. Efficientnet-edgetpu: Creating accelerator-optimized neural networks with automl.

[25] Hariharan, B., Arbeláez, P., Girshick, R., and Malik, J. Simultaneous detection and segmentation. In *Computer Vision – ECCV 2014* (Cham, 2014), D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., Springer International Publishing, pp. 297–312.

[26] Hatamizadeh, A., Yin, H., Heinrich, G., Kautz, J., and Molchanov, P. Global context vision transformers, 2023.

[27] He, K., Gkioxari, G., Dollár, P., and Girshick, R. B. Mask R-CNN. *CoRR abs/1703.06870* (2017).

[28] Hegde, N., Hipp, J. D., Liu, Y., Emmert-Buck, M., Reif, E., Smilkov, D., Terry, M., Cai, C. J., Amin, M. B., Mermel, C. H., Nelson, P. Q., Peng, L. H., Corrado, G. S., and Stumpe, M. C. Similar image search for histopathology: Smily. *npj Digital Medicine 2*, 1 (June 2019).

[29] Hendrycks, D., and Gimpel, K. Gaussian error linear units (gelus), 2023.

[30] Hu, J., Shen, L., and Sun, G. Squeeze-and-excitation networks. *CoRR abs/1709.01507* (2017).

[31] Iqbal, S., and Qureshi, A. N. A heteromorphous deep cnn framework for medical image segmentation using local binary pattern. *IEEE Access* (2022).

[32] Iqbal, S., Qureshi, A. N., Alhussein, M., Choudhry, I. A., Aurangzeb, K., and Khan, T. M. Fusion of textural and visual information for medical image modality retrieval using deep learning-based feature engineering. *IEEE Access* (2023).

[33] Jaccard, P. THE DISTRIBUTION OF THE FLORA IN THE ALPINE ZONE.1. *New Phytologist 11*, 2 (Feb. 1912), 37–50.

[34] Jaccard, Paul. Étude comparative de la distribution florale dans une portion des alpes et du jura.

[35] Jung, H., and Oh, Y. Towards better explanations of class activation mapping. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (October 2021), pp. 1336–1344.

[36] Kather, J. N., Weis, C.-A., Bianconi, F., Melchers, S. M., Schad, L. R., Gaiser, T., Marx, A., and Zöllner, F. G. Multi-class texture analysis in colorectal cancer histology. *Scientific Reports 6*, 1 (2016).

[37] Kingma, D. P., and Ba, J. Adam: A method for stochastic optimization, 2017.

[38] Kumar, A., Kim, J., Cai, W., Fulham, M., and Feng, D. Content-based medical image retrieval: A survey of applications to multidimensional and multimodality data. *Journal of Digital Imaging 26*, 6 (July 2013), 1025–1039.

[39] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. Backpropagation applied to handwritten zip code recognition. *Neural Computation 1*, 4 (1989), 541–551.

[40] Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE 86*, 11 (1998), 2278–2324.

[41] Li, Z., Zhang, X., Müller, H., and Zhang, S. Large-scale retrieval for medical image analytics: A comprehensive review. *Medical Image Analysis 43* (2018), 66–84.

[42] Lin, T., Maire, M., Belongie, S. J., Bourdev, L. D., Girshick, R. B., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft COCO: common objects in context. *CoRR abs/1405.0312* (2014).

[43] Lin, T.-Y., Dollar, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (July 2017).

[44] Liu, S., Qi, L., Qin, H., Shi, J., and Jia, J. Path aggregation network for instance segmentation. *CoRR abs/1803.01534* (2018).

[45] Liu, W., Rabinovich, A., and Berg, A. C. Parsenet: Looking wider to see better. *CoRR abs/1506.04579* (2015).

[46] Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., and Xie, S. A convnet for the 2020s, 2022.

[47] McInnes, L., Healy, J., and Melville, J. Umap: Uniform manifold approximation and projection for dimension reduction, 2020.

[48] Minaee, S., Boykov, Y., Porikli, F., Plaza, A., Kehtarnavaz, N., and Terzopoulos, D. Image segmentation using deep learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence 44*, 7 (2022), 3523–3542.

[49] Mohammad Alizadeh, S., Sadegh Helfroush, M., and Müller, H. A novel siamese deep hashing model for histopathology image retrieval. *Expert Systems with Applications 225* (2023), 120169.

[50] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems* (2019), H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32, Curran Associates, Inc.

[51] Radenovic, F., Iscen, A., Tolias, G., Avrithis, Y., and Chum, O. Revisiting oxford and paris: Large-scale image retrieval benchmarking. *CoRR abs/1803.11285* (2018).

[52] Radenović, F., Tolias, G., and Chum, O. Fine-tuning cnn image retrieval with no human annotation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2019).

[53] Rahaman, M. M., Li, C., Wu, X., Yao, Y., Hu, Z., Jiang, T., Li, X., and Qi, S. A survey for cervical cytopathology image analysis using deep learning. *IEEE Access* (2020).

[54] Selvaraju, R. R., Das, A., Vedantam, R., Cogswell, M., Parikh, D., and Batra, D. Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization. *CoRR abs/1610.02391* (2016).

[55] Shafiei, S., Babaie, M., Kalra, S., and Tizhoosh, H. R. Colored kimia path24 dataset: Configurations and benchmarks with deep embeddings, 2021.

[56] Shao, S., Chen, K., Karpur, A., Cui, Q., Araujo, A., and Cao, B. Global features are all you need for image retrieval and reranking, 2023.

[57] Shao, S., Chen, K., Karpur, A., Cui, Q., Araujo, A., and Cao, B. Global features are all you need for image retrieval and reranking. In *2023 IEEE/CVF Int. Conf. on Computer Vision (ICCV)* (2023).

[58] Sikaroudi, M., Hosseini, M., Gonzalez, R., Rahnamayan, S., and Tizhoosh, H. R. Generalization of vision pre-trained models for histopathology. *Scientific Reports 13*, 1 (2023).

[59] Singh, B., and Davis, L. S. An analysis of scale invariance in object detection - SNIP. *CoRR abs/1711.08189* (2017).

[60] Spanhol, F. A., Oliveira, L. S., Petitjean, C., and Heutte, L. A dataset for breast cancer histopathological image classification. *IEEE Transactions on Biomedical Engineering* (2016).

[61] Tabatabaei, Z., Colomer, A., Moll, J. O., and Naranjo, V. Toward more transparent and accurate cancer diagnosis with an unsupervised cae approach. *IEEE Access 11* (2023), 143387–143401.

[62] Tabatabaei, Z., Colomer, A., Moll, J. O., and Naranjo, V. Siamese content-based search engine for a more transparent skin and breast cancer diagnosis through histological imaging, 2024.

[63] Tan, M., and Le, Q. V. Efficientnetv2: Smaller models and faster training, 2021.

[64] Tan, M., Pang, R., and Le, Q. V. Efficientdet: Scalable and efficient object detection. *CoRR abs/1911.09070* (2019).

[65] Tan, M., Pang, R., and Le, Q. V. Efficientdet: Scalable and efficient object detection, 2020.

[66] Teichmann, M., Araújo, A., Zhu, M., and Sim, J. Detect-to-retrieve: Efficient regional aggregation for image search. *CoRR abs/1812.01584* (2018).

[67] Viola, P., and Jones, M. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001* (2001), vol. 1, pp. I–I.

[68] Viola, P., and Jones, M. J. Robust real-time face detection. *International Journal of Computer Vision 57*, 2 (May 2004), 137–154.

[69] Wang, H., Du, M., Yang, F., and Zhang, Z. Score-cam: Improved visual explanations via score-weighted class activation mapping. *CoRR abs/1910.01279* (2019).

[70] Wang, H., Wang, Y., Zhou, Z., Ji, X., Gong, D., Zhou, J., Li, Z., and Liu, W. Cosface: Large margin cosine loss for deep face recognition. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2018), pp. 5265–5274.

[71] Wang, H., Wang, Y., Zhou, Z., Ji, X., Li, Z., Gong, D., Zhou, J., and Liu, W. Cosface: Large margin cosine loss for deep face recognition. *CoRR abs/1801.09414* (2018).

[72] Wang, W., Dai, J., Chen, Z., Huang, Z., Li, Z., Zhu, X., Hu, X., Lu, T., Lu, L., Li, H., Wang, X., and Qiao, Y. Internimage: Exploring large-scale vision foundation models with deformable convolutions, 2023.

[73] Woo, S., Debnath, S., Hu, R., Chen, X., Liu, Z., Kweon, I. S., and Xie, S. Convnext v2: Co-designing and scaling convnets with masked autoencoders, 2023.

[74] Yang, P., Zhai, Y., Li, L., Lv, H., Wang, J., Zhu, C., and Jiang, R. A deep metric learning approach for histopathological

image retrieval. *Methods 179* (2020), 14–25. Interpretable machine learning in bioinformatics.

[75] Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. How transferable are features in deep neural networks? *CoRR abs/1411.1792* (2014).

[76] Zaidi, S. S. A., Ansari, M. S., Aslam, A., Kanwal, N., Asghar, M., and Lee, B. A survey of modern deep learning based object detection models. *Digital Signal Processing 126* (June 2022), 103514.

[77] Zeiler, M. D., and Fergus, R. Visualizing and understanding convolutional networks. *CoRR abs/1311.2901* (2013).

[78] Zhou, B., Khosla, A., Lapedriza, À., Oliva, A., and Torralba, A. Learning deep features for discriminative localization. *CoRR abs/1512.04150* (2015).

[79] Zhou, Y.-T., and Chellappa, R. Computation of optical flow using a neural network. *IEEE 1988 International Conference on Neural Networks* (1988), 71–78 vol.2.

[80] Zoph, B., and Le, Q. V. Neural architecture search with reinforcement learning. *CoRR abs/1611.01578* (2016).

[81] Zoph, B., Vasudevan, V., Shlens, J., and Le, Q. V. Learning transferable architectures for scalable image recognition. *CoRR abs/1707.07012* (2017).

[82] Zou, Z., Chen, K., Shi, Z., Guo, Y., and Ye, J. Object detection in 20 years: A survey. *Proceedings of the IEEE 111*, 3 (Mar. 2023), 257–276.