

Vol. 33, No. 1, 2024

(this volume contains accepted papers online and is not closed yet)

Machine
GRAPHICS & VISION

International Journal

Published by
The Institute of Information Technology
Warsaw University of Life Sciences – SGGW
Nowoursynowska 159, 02-776 Warsaw, Poland

in cooperation with
The Association for Image Processing, Poland – TPO

AN IMPROVED GENERATIVE DESIGN APPROACH BASED ON GRAPH GRAMMAR FOR PATTERN DRAWING

Yufeng Liu*, Yangchen Zhou, Fan Yang^{ORCID}, Song Li and Jun Wu^{ORCID}

College of Information Engineering, Nanjing University of Finance and Economics, Nanjing, China

**Corresponding author: Yufeng Liu (yfengliu28@126.com)*

Abstract Generative design is used to efficiently generate design solutions with powerful computational methods. Generative design based on shape grammar is currently the most commonly used approach, but it is difficult for shape grammar to formally analyze the generated pattern. Graph grammar derived from one-dimensional character grammar is mainly used for generating and analyzing abstract models of visual languages. However, there is a significant gap between the generated node-edge graphs and the representation of shape appearance. To address these problems, we propose an improved generative design approach based on virtual-node based continuous Coordinate Graph Grammar (vcCGG). This approach defines a new type of grammatical rule named node transformation rules to convert nodes into shapes with node transformation applications. By combining node transformation applications and L-applications in vcCGG, we can generate a node-edge graph as the structure of the pattern through L-applications, and then draw the shape outline, next adjust the positions of these shapes, thus relating abstract structures and the physical layouts of visual languages. At the end of the paper, we provide an example application of this approach: generating an illustration from Emma Talbot using a combination of node transformation applications and L-applications.

Keywords: generative design; graph grammar; shape grammar; node transformation rules; pattern drawing.

1. Introduction

Design is a complex solution process that involves professional knowledge, innovative ability, comprehensive experience, aesthetic literacy, and use of scientific technology. With the rapid development and popularization of new intelligent design automation technologies such as machine learning, additive manufacturing, artificial intelligence, and cloud computing, design approaches are constantly expanding. As a developing design approach, generative design has been extensively studied in academia. Since the introduction of generative design based on shape grammar, as proposed by G. Stiny and J. Gips in 1971 [18], generative design has been introduced into different fields such as architectural design [5], product customization design [9], and visual communication design [14].

Shape grammar is a generation system oriented toward design. It is a design inference approach based on rules, using simple shapes as basic elements to establish the rules for the generation of complex shapes. The foundational rules involve spatial transformations such as translation, scaling, rotation and mirroring, which make one shape part of another shape. With limited predefined rules, there can be an infinite number of

designs generated through shape grammar. Following predefined rules, shape grammar can iteratively replace shapes to generate various patterns. However, shape grammar can generate only the shapes that consist of simple shapes such as lines, points and rectangles. Therefore, it is not yet widely used in computer-aided architectural design (CAAD) applications. Most designers design buildings manually or semi-automatically on CAD platforms, e.g. Revit and AutoCAD.

Shape grammar focuses on generative design, while graph grammar derived from one-dimensional character grammar focuses on modeling and analyzing the syntax and semantics of visual languages. Shape grammar supports only unidirectional workflows. It takes the initial shape and transformation rules as inputs to generate a preliminary design and then adjusts the preliminary design by the rules to generate the final design. In contrast, graph grammars have a bidirectional workflow across derivation and specification. Similarly, the graph grammar derivation process derives graphs by repeatedly applying given productions. The graph grammar reduction process, on the other hand, takes graphs and productions as inputs to parse the graphs by applying productions in a bottom-up fashion. However, there is a significant gap between the generated node-edge graphs and the representation of shape appearance for graph grammar.

In our previous work, we proposed an enhanced grammar system for shape generation [12]. This system defines shape rules to transform edges into shapes by shape applications, which builds an inherent relation between abstract structures and physical layouts of visual languages. The main weakness of this system is the position invariance that reduces the flexibility of design. To address the aforementioned issue, our research focuses on an analysis of semantic relations among shapes that make up a pattern. We propose a generative design approach based on vCGG (virtual-node based Coordinate Graph Grammar) [10]. Our approach defines a new type of grammatical rule named node transformation rules to convert nodes into shapes with node transformation applications. By combining node transformation applications and L-applications in vCGG, we can generate a node-edge graph as the structure of the pattern through L-applications, and then draw the outlines of shapes with node transformation applications, next adjusting the positions of these shapes.

In summary, this paper presents an improved generative design approach that automatically generates or validates patterns conforming to the specified rules. First, the structure of the target pattern is generated through vCGG, and then the nodes are converted into shapes according to the node transformation rules. Finally, the position of the shape is adjusted based on the edge attributes, and the target pattern is generated. This approach can set L-applications and node transformation rules in advance for drawing patterns, and can also formally validate a target pattern to determine whether it belongs to the pattern generated by the specified rules.

This paper addresses the aforementioned problems and makes the following contributions:

- An improved approach for grammar specification, grammar induction, generation and validation of pattern based on the vCGG formalism.
- A complete graph grammar for specifying and analyzing patterns that are composed of multiple geometric shapes.
- According to the concrete requirements, productions and transformation rules are designed to achieve customized designs.

The rest of this paper is organized as follows. Section 2 reviews the related works, including patterns generated by shape grammars, several typical graph grammars and our approach. Section 3 introduces the approach framework, including vCGG and node transformation rules. Next, Section 4 gives an example of the Cloud & Bunny rabbit pattern from Emma Talbot. Section 5 compares our approach and other generative design approaches. Finally, Section 6 concludes the paper and mentions future work.

2. Related works

In 1971, G. Stiny and J. Gips proposed that shape grammar is a generative system oriented toward design. G. Stiny detailed the concept and entire application process of shape grammar in 1980 [17]. Design based on shape grammar was first applied in the field of architectural design. M. Agarwal and J. Cagan [1] proposed the coffee machine shape grammar as the first application of shape grammar in product design, demonstrating its use for generating single products before gradually being applied to product design more broadly. The coffee machine grammar is a parametric grammar consisting of 100 manually created rules and labeled two-dimensional shape grammar implemented through a Java-based application program. Its objective is to provide designers with selectable design inspirations during the conceptual exploration phase. However, this method has limitations because its conceptual nature lacks practical production benefits, resulting in visual operational difficulties due to numerous labels.

H. H. Chau [3] concluded, through analysis of various electronic and fast-moving consumer products, that the appearance of these products is largely determined by straight lines, arcs, and their orthogonal projections. M. Pugliese and J. Cagan [13] summarized previous research methods and found that grammar has become a design tool for creating structures and functional requirements. However, there is no specific method for establishing and maintaining product brand characteristics in the field of product generation design. The field faces two challenges: engineers and designers need tools to help understand, express, and maintain product brands, and engineers, designers, and brand strategists need a common platform to discuss product brands. X. Chen et al. [4] focused on geometric shape in packaging design, proposing an application of shape grammar for packaging design research with personal care bottles as an example in experimentation. S. Wannarumon et al. [20] proposed a method for generating jewelry designs using shape grammar to support designers in exploring shapes as inspiration

sources with ring design as practice examples. S. Garcia and L.Romao [7] coded various types embedded in multifunctional chair classes to develop generative design tools usable during the chair concept design stage. Y. Yu et al. [21] proposed a method of generating origami pattern based on shape grammar recursive applications of shape rewriting rules. In addition, shape grammar provides a perspective and modeling technique for creating origami tessellation patterns.

Compared to shape grammar in the field of design, graph grammar has the characteristics of automated generation and specification. Designers can explore different design options by defining symbols, rules, and parameters, quickly generate a large number of design schemes, and make adjustments and modifications when necessary to improve design efficiency and innovation. H. Bunke [2] proposed attributed programmed graph grammars as a generative tool in image understanding. Based on that, an image understanding system was built to extract descriptions from input images, where a system consists of two major subsystems for preprocessing and segmentation, and understanding, respectively. H. Göttler et al. [8] described the data structures in terms of attributed graphs and their changes in terms of attributed graph productions in an object-oriented manner, applying Graph Grammar to CAD systems.

In the field of architectural design, X. Wang et al. [19] presented a generic approach for grammar specification, grammar induction, validation, and design generation of house floor plans using their path graphs based on the reserved graph grammar formalism (RGG). This approach validates floor plans in different styles with user-specified graph productions and the derivation process is capable of generating floor plan designs. G. Ślusarczyk [23] proposed a framework for supporting the design process by defining design requirements over graph-based representations of designs. First, hierarchical layout graph grammars are used to generate hierarchical layout hypergraphs (HL-graphs) that represent designs; then, local and global graph requirements are defined over HL-graphs, which correspond to design constraints. The proposed ontological interpretations transform first-order and monadic second-order logic formulas expressing design criteria into equivalent local and global graph requirements. The satisfiability of graph requirements by representations of designs allows for checking correctness of design solutions. In subsequent research, G. Ślusarczyk et al. [24] proposed CP-graph grammars to support building layout design, where the grammar rules are combined with semantic-driven embedding transformations and the derivations in this type of grammars are defined. The possibility of relating attributes of right-hand sides to that of the left-hand sides enables the system to capture parametric modelling knowledge. The proposed generative method allows the system to automatically model alternative floor layouts with similar structures but different geometry and parameters, which can be easily adapted to different use case scenarios and environmental conditions.

Apart from the architectural design, graph grammar has been applied to different

fields, including mechanical parts description [6], XML validation [16], cluster analysis [22], entity-relationship (E-R) diagram validation [11], and Web pattern recognition and validation [15]. Overall, graph grammar is a powerful tool for defining and validating graph models, hence the generative design method in this paper is proposed within the framework of graph grammar.

Because patterns are composed of various styles of shapes, there is a positional correlation between each shape. The structure of patterns is generated through graph grammar, which abstracts the positional relationships between various shapes. Then we convert the node-edge graph generated by graph grammar into shapes through node transformation rules, enabling graph grammar to generate shapes and draw patterns. Moreover, graph grammar parsing can check whether a target pattern belongs to the pattern set defined by the rules.

3. Improved generative design approach framework

VCGG is divided into virtual-node based discrete Coordinate Graph Grammar (vdCGG) and virtual-node based continuous Coordinate Graph Grammar (vcCGG) based on different granularity descriptions of spatial semantics. Due to the strict coordinate matching mechanism required in this approach, we choose vcCGG as the basic framework. Below is the theoretical framework of the improved approach.

Definition 3.1. A *directed graph* G on a given label set L is a 2-tuple (N, E) . L consists of a virtual label set L_v and a real label set L_r , where L_r consists of a non-terminal label set L_{NT} and a terminal label set L_T . N is a node set and consists of a virtual node set N_V and a real node set N_r , where N_r consists of a nonterminal node set N_{NT} and a terminal node set N_T . E is a directed edge set.

Mapping for G includes the following:

- $f_{NL} : N \rightarrow L$ is a mapping that assigns a label $l \in L$ to node $n \in N$;
- $f_{NC} : N \rightarrow R \times R$ is a mapping that assigns a 2D coordinate $c \in R \times R$ to node $n \in N$;
- $f_{EN_s} : E \rightarrow N$ is a mapping that assigns the start node to directed edge $e \in E$;
- $f_{EN_e} : E \rightarrow N$ is a mapping that assigns the end node to directed edge $e \in E$.

Definition 3.2. A *production* $p: G_L := G_R$ is made up of a left-hand-side (or left graph) G_L and a right-hand-side (or right graph) G_R . For a production, there exists a bijection $f_{NN} : G_L.N_v \leftrightarrow G_R.N_v$ between $N_v \in G_L$ and $N_v \in G_R$, where $G_L.N_v$ is a virtual node set N_v of G_L and $G_R.N_v$ is a virtual node set N_v of G_R .

A production also satisfies the following conditions:

- $\forall n((n \in G_L.N_v) \Rightarrow (f_{NC}(n) = f'_{NC}(f_{NN}(n))))$, where f_{NC} is a mapping that assigns a coordinate to node $n \in G_L$ and f'_{NC} is a mapping that assigns a coordinate to $n \in G_R$;

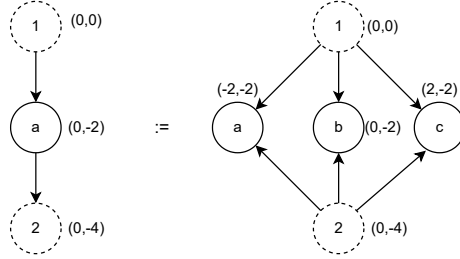


Fig. 1. vcCGG production.

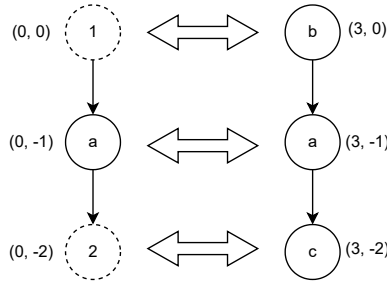


Fig. 2. The isomorphic graphs in vcCGG.

- $\forall n((n \in G_L.N_v) \Rightarrow (f_{NL}(n) = f'_{NL}(f_{NN}(n))))$, where f_{NL} is a mapping that assigns a label to node $n \in G_L$ and f'_{NL} is a mapping that assigns a label to $n \in G_R$;
- $\forall n_1, n_2((n_1, n_2 \in G_L.N_v) \wedge (n_1 \neq n_2) \Rightarrow (f_{NL}(n_1) \neq f_{NL}(n_2)))$;
- $\forall n_1, n_2((n_1, n_2 \in G_R.N_v) \wedge (n_1 \neq n_2) \Rightarrow (f'_{NL}(n_1) \neq f'_{NL}(n_2)))$.

VcCGG stipulates that there is a bijection between the virtual node sets at G_L and G_R , and the corresponding nodes have the same labels and coordinates. In addition, to avoid ambiguity during graph embedding, each virtual node in the same graph must have a unique label, which can be represented by a unique integer.

For example, Fig. 1 is a legal vcCGG production, where the dashed circle represents the virtual nodes and the solid circle represents the real nodes. There is a bijection between the left and right graphs of the production, and the corresponding nodes have the same labels '1', '2' and equal coordinates (0, 0) and (0, 4).

Definition 3.3. Let G and Q be directed graphs. G and Q are **isomorphic**, denoted as $G \approx Q$, if and only if the following conditions hold:

- There exists a bijection between the nodes of G and Q , namely, $f_{NN} : G.N \leftrightarrow Q.N$;

- There exists a bijection between the edges of G and Q , namely, $f_{EE} : G.E \leftrightarrow Q.E$;
- $\forall n((n \in G.N) \vee (n \in Q.N) \Rightarrow (f_{NL}(n) \in L_v) \vee (f'_{NL}(f_{NN}(n)) \in L_v) \vee (f_{NL}(n) = f'_{NL}(f_{NN}(n))))$, where f_{NL} is a mapping that assigns a label to node $n \in G$; f'_{NL} is a mapping that assigns a label to $n \in Q$;
- $\forall e((e \in G.E) \vee (e \in Q.E) \Rightarrow (f_{NN}(f_{EN_s}(e)) = f_{EN_s}(f_{EE}(e))))$;
- $\forall e((e \in G.E) \vee (e \in Q.E) \Rightarrow (f_{NN}(f_{EN_e}(e)) = f_{EN_e}(f_{EE}(e))))$.

When determining whether a pair of graphs satisfies the isomorphic condition, virtual nodes have a higher abstract degree than real nodes and can match any labeled node. Fig. 2 is an example of graph isomorphism in vcCGG, where all nodes and edges satisfy a bijective relationship. Real node ‘a’ and the corresponding nodes must have the same label, while virtual nodes ‘1’ and ‘2’ can match any labeled node. In Fig. 2, node ‘1’ matches ‘b’ and node ‘2’ matches node ‘e’.

Definition 3.4. Let G be a directed graph referred to as the host graph and Q be the subgraph of G . Let $G_{L|R}$ be the left or the right hand-side of a production. Q is called a **redex** of G with respect to $G_{L|R}$, denoted as $Q \in \text{redex}(G, G_{L|R})$ if and only if the following conditions hold:

- $Q \approx G_{L|R}$;
- $\forall n((n \in Q.N \wedge ((f'_{NL}(f_{NN}(n)) \in L_r) \Rightarrow (d_s(n) = d_s(f_{NN}(n))) \wedge (d_e(n) = d_e(f_{NN}(n))))))$;
- $\forall n_1, n_2((n_1, n_2 \in Q.N) \Rightarrow (f_{NC}(n_1) - f_{NC}(n_2) = f'_{NC}(f_{NN}(n_1)) - f'_{NC}(f_{NN}(n_2))))$.

The nodes of a redex could be divided into two types: the nodes matched by the virtual nodes (context nodes) of the production, and the nodes matched by the non-virtual nodes (inner nodes) of the production. All the edges between the redex and the rest host graph are only allowed to be connected with the former type of nodes.

Definition 3.5. A **L/R application** to graph G is a process that generates graph G' using production $p: G_L := G_R$, denoted as $G \rightarrow^p G'$ (L -application) or $G \rightarrow^p G'$ (R -application).

The L-application in vcCGG is as follows:

1. Generate an instance of the production as a copy of the production.
2. Translate the coordinates of the instance’s G_R by the offset between any matched nodes in the redex Q and G_L .
3. Delete edges in the redex Q and nodes that match the real nodes in G_L from the host graph.
4. According to the mapping between the virtual node of G_L and the redex Q , glue the virtual node of G_R to the corresponding node in the redex Q and remove the virtual label from the host graph.

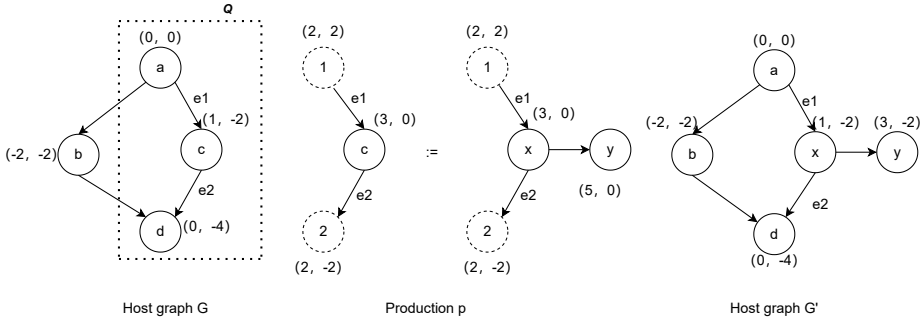


Fig. 3. New host graphs generated by a production.

Fig. 3 depicts an L-application process that generates new host graph G' using production p: $G_L := G_R$.

1. Generate an instance of production p.
2. Find a redex of G with respect to G_L : In the host graph G, we denote a graph in the dashed box as graph Q. $Q \approx G_L$ and the coordinate differences of the corresponding nodes are (2, 2), so $Q \in \text{redex}(G, G_L|_R)$.
3. Subtract all node coordinates of G_R (2, 2).
4. Delete edge 'e1', 'e2' and node 'c' from G.
5. Glue virtual node '1' of G_R to real node 'a' of G and virtual node '2' of G_R to real node 'd' of G; and remove the virtual label from the host graph.

Definition 3.6. A node transformation rule is a 4-tuple $(cset, cpoint, ops, parm)$, where

- *cset* is a set of coordinates as the points to represent a shape;
- *cpoint* is the mean point of *cset*;
- *ops* is the operations performed on the *cset*, such as translation, rotation, scaling, etc.;
- *parm* is the parameter of the *ops*, such as the offset of translation or the angle of rotation.

Given a node transformation rule, the node transformation application is a process that draws the outline of a shape from the perspective of the user using node transformation rules. Below are the steps for a node transformation application:

1. Draw a shape based on the outline described by a node's *cset*, and make the *cpoint* coincide with the node. As shown in Fig. 4, a node transformation rule is to transform a node into a rectangle. Use this node transformation rule for node A and B: make the *cpoint* of this rectangle coincide with node A and B, and transform edge e_1 connecting A and B to line segment l_1 ;

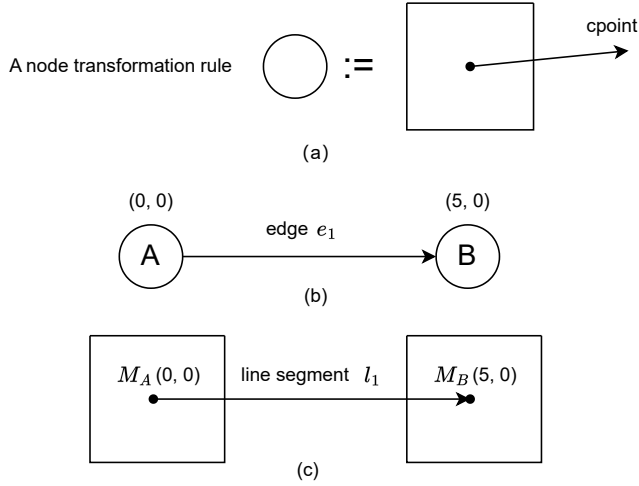


Fig. 4. Demonstration figure of step 1.

2. As shown in Fig. 5, deform the shape by the following methods according to ops and parm:

(a) Translation: Let A be a shape, and the position of A can change along the X and Y axes, i.e.,

$$\forall (x, y) \in A, (x', y') = (x + a, y + b),$$

where a is the distance that the position of A changes on the X axes and b is the distance that the position of A changes on the Y axes.

(b) Scale: Let A be a shape that can expand or shrink in a certain proportion, i.e.,

$$\forall (x, y) \in A, \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} S & 0 \\ 0 & S \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}, \text{ where } S \text{ is the factor by which shape } A \text{ expands or shrinks.}$$

(c) Stretch: Let A be a shape that can be elongated or shortened along the X and Y axes. Specifically, if the factors of elongation or shortening along the X and Y axes are equal, A can be considered to be scaled, i.e.,

$$\forall (x, y) \in A, \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} Sx & 0 \\ 0 & Sy \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix},$$

where Sx is the factor by which A is elongated or shortened along the X axes and Sy is the factor by which A is elongated or shortened along the Y axes.

(d) Rotate: Let A be a shape that can rotate θ ($0 < \theta < 2\pi$) counterclockwise around the cpoint $M_A(X_A, Y_A)$, i.e.,

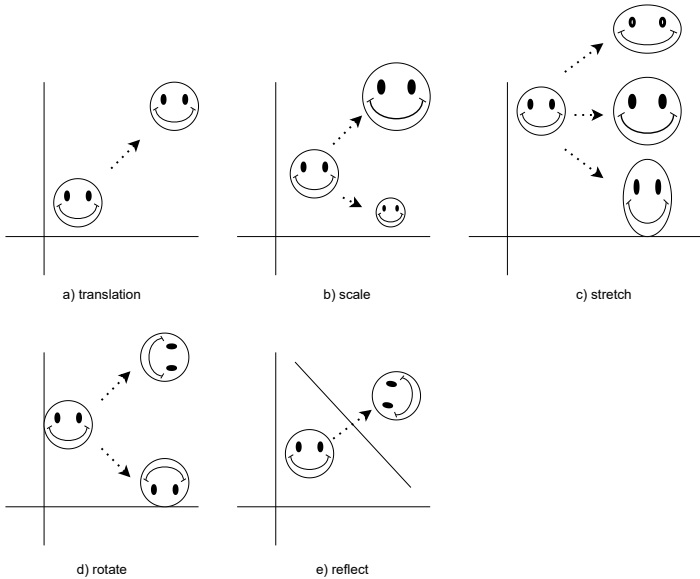


Fig. 5. A new shape formed by 5 operations.

$$\forall (x, y) \in A, (x', y') = ((X_A - x) \cos\theta - (Y_A - y) \sin\theta + X_A, (X_A - x) \sin\theta + (Y_A - y) \cos\theta + Y_A).$$

(e) Reflect: Let A be a shape. $\forall l : PX + QY + M = 0 (P^2 + Q^2 > 0)$, new shape A' is a mirror image of A across line l , i.e.,

$$\forall (x, y) \in A, (x', y') = \left(x - \frac{2P(Px + Qy + M)}{P^2 + Q^2}, y - \frac{2Q(Px + Qy + M)}{P^2 + Q^2}\right).$$

3. Render the shape from the user's perspective based on the outline described by the cset through its own operations.
4. Adjust the position of the shape based on the attributes of the line segment l_1 .

Definition 3.7. For shape A and shape B , A and B are **separated** if and only if $\exists l : Px + Qy + M = 0 (P^2 + Q^2 > 0)$, A and B are on both sides of line l , as shown in Fig. 6.

As shown in Fig. 7, for shape A and B , M_A is the cpoint of A and M_B is the cpoint of B . M_A and M_B are connected through a directed line segment l_{AB} , where M_A is the start point of l_{AB} and M_B is the end point of l_{AB} . The position of M_A will change according to the attribute of l_{AB} , and the position of A will be changed following the changes in M_A position. The attribute of l_{AB} is 'far from d ' or 'near d ', where d is the distance at which the M_A position changes. When using node transformation rules to transform

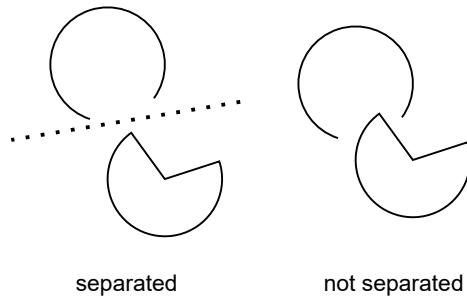


Fig. 6. The two shapes are separated or not.

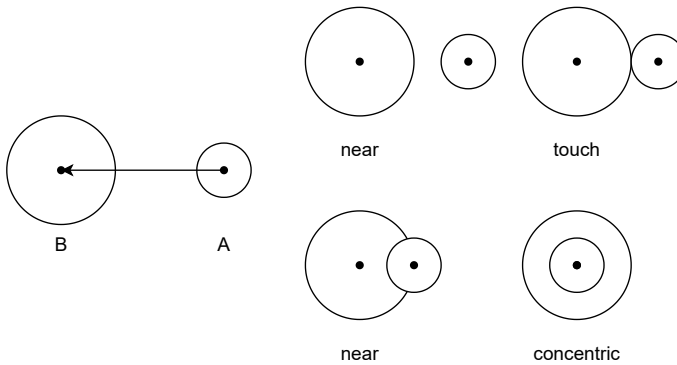


Fig. 7. A is near B; A touches B; A is concentric to B.

node A and B into shape A and B, it is necessary to ensure that they are separated. Therefore, if the attribute of l_{AB} is ‘far from d ’, regardless of the value of d , A and B are still separated. So, we won’t limit the value of d when the attribute of l_{AB} is ‘far from d ’.

Definition 3.8. *If the attribute of l_{AB} is ‘near d ’, A may touch B or be concentric with B during the process of changing the position of A.*

- *Touch: $A.cset \cap B.cset \neq \emptyset$ for the first time;*
- *Concentric: M_A coincides with M_B .*

For convenience, when users want A to touch B or be concentric with B, they can

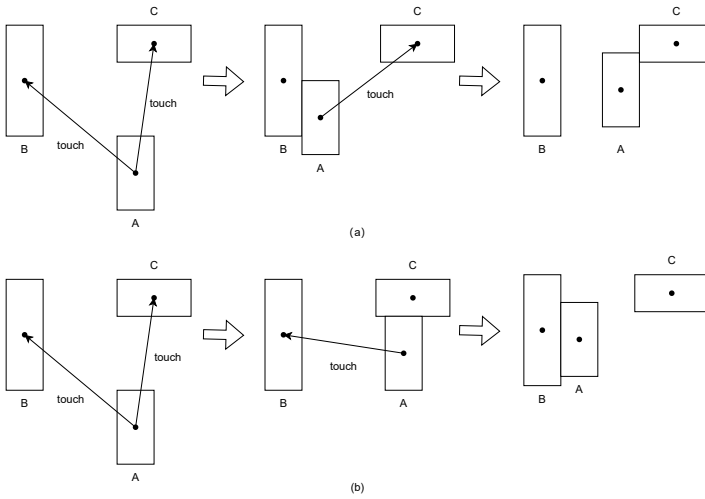


Fig. 8. The final position of A will change due to the order of touch B or C.

set the l_{AB} attribute to 'touch' or 'concentric'. Before the position of M_A changes, make $D_{\max} = |M_A - M_B|$. So, $0 < d \leq D_{\max}$ when the attribute of l_{AB} is 'near d '.

As shown in Fig. 8, for shape A, when M_A is the starting point of two or more directed line segments, the position of A must be changed at least twice, and different changing sequences can lead to different positions. As shown in the Fig. 8, A needs to touch both B and C, and the final position of A will change based on the order of it touches B or C. Therefore:

- When the X coordinate of the end nodes is different, the position of start node first changes toward the end node with a smaller X coordinate;
- When the X coordinate of the end nodes is the same, the position of start node first changes toward the end node with a smaller Y coordinate.

4. An example on rabbit pattern

This section gives an example to illustrating an application of the improved approach, where a set of designed productions and node transformation rules are used to generate a section of the Cloud & Bunny rabbit pattern from Emma Talbot. Emma is passionate about mixed media research and enjoys using various media to create textures, patterns, and collages to integrate into her artistic creations. The Cloud & Bunny rabbit pattern is composed of simple geometric shapes such as arcs, rectangles, triangles, etc., forming

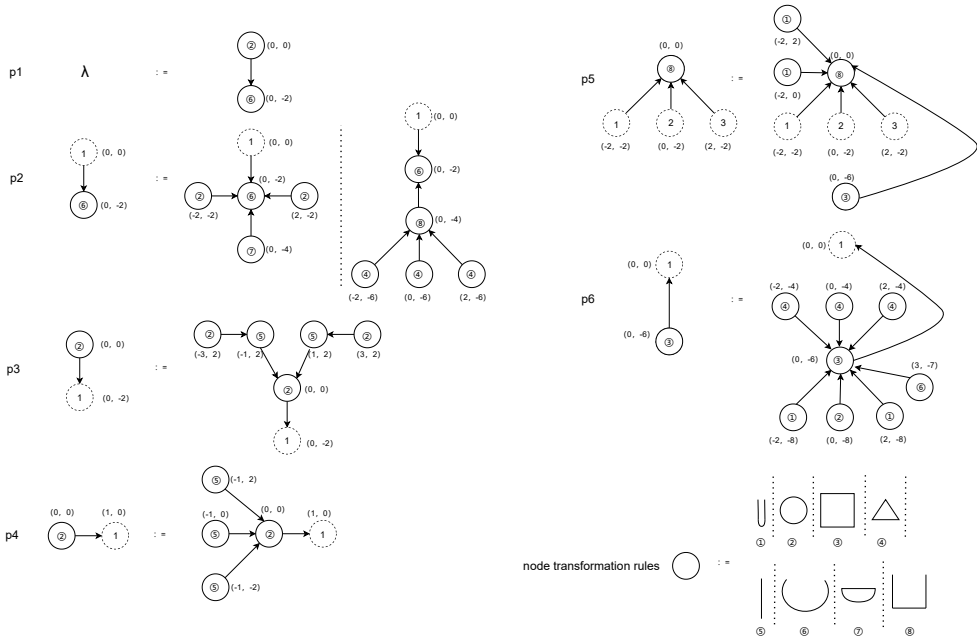


Fig. 9. Productions for a bunny rabbit.

patterns of rabbits, flowers, and clouds. In this paper, a rabbit pattern is selected as the generated pattern. Fig. 9 shows a set of vcCGG productions and eight node transformation rules as a grammar set for the rabbit pattern, where the vcCGG productions are used for the abstract models of pattern and node transformation rules describe physical layouts. For the vcCGG productions, the initial symbol ‘ λ ’ denotes the beginning of graph grammar. ‘ λ ’ is used to generate the right graph of p1 through production p1 and then generate the target structure of the pattern based on the remaining productions p2-p6. For the productions in Fig. 9, virtual nodes, which are represented by a dashed circle and labeled ‘1’, ‘2’, and ‘3’, are used to match coordinates; real nodes, which are represented by a solid circle and labelled ‘①’, ‘②’, and ‘③’, are converted into shapes. For the node transformation rules in Fig. 9, we set eight shapes to generate the final pattern, including circle, rectangle, triangle, etc.

Fig. 10 shows a process of generating a rabbit pattern using the productions and node transformation rules above. When using an L-application to generate the structure of the target pattern, an attribute is assigned to each generated edge. The attribute can be ‘near’, ‘touch’ or ‘concentric’. If the attribute is ‘near’, the distance needs to be

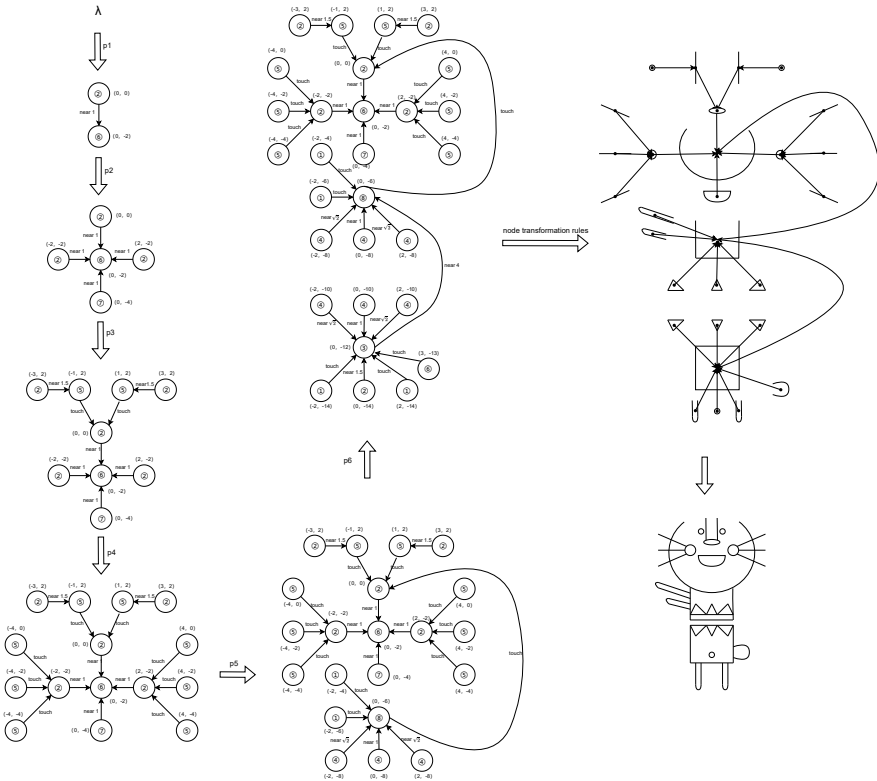


Fig. 10. Generation of a bunny rabbit.

given as parameter. When using node transformation rules for the final generated node-edge graph, each node is traversed and converted into a shape based on the associated label. Then, each edge is traversed, the position of each shape is adjusted based on the attribute of each edge, and the target pattern is ultimately obtained.

5. Comparisons with other generative design approaches

In this section, we compare our approach proposed in this paper with shape grammar, edge transformation grammar [12] and CP-graph grammar [24]. Shape grammar is a design inference approach based on rules, using simple shapes as basic elements to establish the rules for the generation of complex shapes. Edge transformation grammar

defines shape rules to transform edges into shapes by shape applications. The CP-graph grammar is used to automatically generate CP-graphs corresponding to new layout designs with non-geometrical properties (like sizes, areas) specified by graph attributes.

As Table 1 shows, these approaches can all design shapes through derivation. However, when drawing patterns using shape grammar, different shapes of a pattern are related only in terms of position and have no semantic relations. Therefore, it is difficult to formally analyze the generated pattern. Our approach based on vcCGG can formally validate a target pattern to determine whether it belongs to the pattern generated by the specified rules by combining node transformation applications and L-applications. Moreover, after designing the transformation rules for shape grammar, edge transformation grammar and CP-graph grammar, they are unable to adjust the size and position of the shape, resulting in a lack of position and size variability. However, for our approach, after generating the structure of the target pattern through vcCGG, the nodes which are converted into shapes according to the node transformation rules can adjust the size and position of themselves. Therefore, in terms of position and size variability, our approach is superior to shape grammar and edge transformation grammar.

Tab. 1. Comparison between approach in this paper, shape grammar, edge transformation grammar and CP-graph grammar.

Approach	Derivation	Parsing	Positional and size variability
Our approach	✓	✓	✓
Shape grammar	✓	×	×
Edge transformation system	✓	✓	×
CP-graph grammar	✓	✓	×

6. Conclusions

When designers use shape grammar to generate patterns, there are no semantic relations among the various shapes that make up the pattern or the small patterns that make up the large patterns. Therefore, it is difficult to formally analyze the generated patterns. In addition, graph grammar is primarily used for generating and analyzing abstract models of visual languages. There is a significant gap between the generated node-edge graphs and the visual representation of shapes, so few researchers have applied these concepts in the design field.

This paper proposes an improved generative design approach for pattern drawing, which introduces node transformation rules in the framework of vcCGG. First, the structure of the target pattern is generated through vcCGG, and then the nodes are converted into shapes according to the node transformation rules. Finally, the position of each

shape is adjusted based on the edge attributes, and the target pattern is generated. In this approach, L-applications and node transformation rules are set in advance for drawing patterns, and a target pattern can be formally analyzed to determine whether it is a pattern generated based on the specified rules.

In the future, we plan to improve the theoretical framework of the improved approach and consider adding gray values to the node transformation rules. If it goes well, we plan to add RGB to it so that the improved approach can be used to design the colored patterns. Moreover, we plan to develop a support system for this approach with a friendly GUI for end users to design graph productions and node transformation rules. The system platform will provide support for grammatical operations and the implementation of related applications.

Acknowledgement

This work was supported by the National Natural Science Foundation of China within the grant No. 62002155 and the National Key Research and Development Program of China within the grant No. 2022YFB3305504.

References

- [1] M. Agarwal and J. Cagan. A blend of different tastes: The language of coffeemakers. *Environment and Planning B: Planning and Design*, 25:205–226, 1998. doi:10.1068/b250205.
- [2] H. Bunke. Graph grammars as a generative tool in image understanding. In: *Graph-Grammars and Their Application to Computer Science*, pp. 8–19, 1983. doi:10.1007/BFb0000096.
- [3] H. H. Chau. *Preserving Brand Identity in Engineering Design Using a Grammatical Approach*. Ph.D. thesis, The University of Leeds, School of Mechanical Engineering, and Keyworth Institute of Manufacturing and Information Systems, 2002. https://www.researchgate.net/publication/286452884_Preserving_brand_identity_in_engineering_design_using_a_grammatical_approach.
- [4] X. Chen, A. McKay, A. de Pennington, and H. H. Chau. Package shape design principles to support brand identity. *Proc. 14th IAPRI World Conference on Packaging*, pp. 1–14, 2004. https://www.researchgate.net/publication/267797410_PACKAGE_SHAPE_DESIGN_PRINCIPLES_TO_SUPPORT_BRAND_IDENTITY.
- [5] G. Díaz, R. F. Herrera, F. Muñoz-La Rivera, and E. Atencio. Generative design for dimensioning of retaining walls. *Mathematics*, 9(16):1918, 2021. doi:10.3390/math9161918.
- [6] M. Flasiński. Use of graph grammars for the description of mechanical parts. *Computer-Aided Design*, 27:403–433, 1995. doi:10.1016/0010-4485(94)00015-6.
- [7] S. Garcia and L. Romao. A design tool for generic multipurpose chair design. In: *Proc. Computer-Aided Architectural Design Futures*, pp. 600–619, 2015. doi:10.1007/978-3-662-47386-3_33.
- [8] H. Göttler, J. Günther, and G. Nieskens. Use graph grammars to design CAD-systems! In: *Graph-Grammars and Their Application to Computer Science*, pp. 396–410, 1990. doi:10.1007/BFb0017402.

- [9] M. Lee, Y. Park, H. Jo, K. Kim, S. Lee, et al. Deep generative tread pattern design framework for efficient conceptual design. *Journal of Mechanical Design*, 144(7):011703, 2022. doi:10.1115/1.4053469.
- [10] Y. Liu and Y. Fan. VCGG: Virtual-node based spatial graph grammar formalism. *Journal of Software*, 32:3669–3683, 2021. doi:10.13328/j.cnki.jos.006164.
- [11] Y. Liu, X.-Q. Zeng, and Y. Zhu. Application of graph grammar EGG to design of ER diagrams. *Computer Engineering and Design*, 2014(3):1071–1075, 2014. <https://api.semanticscholar.org/CorpusID:63040768>.
- [12] Y. Liu, Y. Zhou, F. Yang, and H. Sun. An enhanced grammatical approach for graph drawing. In: *Conf. International Conference on Artificial Intelligence, Virtual Reality, and Visualization AIVRV 2022*, p. 1258803, 2023. doi:10.1117/12.2667201.
- [13] M. Pugliese and J. Cagan. Capturing a rebel: modeling the Harley-Davidson brand through a motorcycle shape grammar. *Research in Engineering Design*, 13:139–156, 2002. doi:10.1007/s00163-002-0013-1.
- [14] C. Qian, R. Tan, and W. Ye. An adaptive artificial neural network-based generative design method for layout designs. *International Journal of Heat and Mass Transfer*, 184:122313, 2022. doi:10.1016/j.jheatmasstransfer.2021.122313.
- [15] A. Roudaki, J. Kong, and K. Zhang. Specification and discovery of web patterns: a graph grammar approach. *Information Sciences*, 328:528–545, 2016. doi:10.1016/j.ins.2015.08.052.
- [16] G. Song and K. Zhang. Visual xml schemas based on reserved graph grammars. In: *Conf. International Conference on Information Technology: Coding and Computing. ITCC 2004*, pp. 687–691, 2004. doi:10.1109/ITCC.2004.1286546.
- [17] G. Stiny. Introduction to shape and shape grammars. *Environment and Planning B: Planning and Design*, 7(3):343–351, 1980. doi:10.1068/b070343.
- [18] G. Stiny and J. Gips. Shape grammars and the generative specification of painting and sculpture. In: *Proc. Conf. International Federation for Information Processing IFIP 1971*, pp. 125–135, 1971. <https://api.semanticscholar.org/CorpusID:36431081>.
- [19] X. Wang, Y. Liu, and K. Zhang. A graph grammar approach to the design and validation of floor plans. *The Computer Journal*, 63:137–150, 2019. doi:10.1093/comjnl/bxz002.
- [20] S. Wannarumon, P. Pradujphonphet, and I. Bohez. An approach of generative design system: Jewelry design application. *IEEE International Conference on Industrial Engineering and Engineering Management*, pp. 1329–1333, 2014. doi:10.1109/IEEM.2013.6962626.
- [21] Y. Yu, T.-C. Hong, A. Economou, and G. Paulino. Rethinking origami: A generative specification of origami patterns with shape grammars. *Computer-Aided Design*, 137:103029, 2021. doi:10.1016/j.cad.2021.103029.
- [22] K.-B. Zhang, M. A. Orgun, and K. Zhang. A prediction-based visual approach for cluster exploration and cluster validation by HOV3. In: *Proc. Knowledge Discovery in Databases. PKDD 2007*, pp. 336–349, 2007. doi:10.1007/978-3-540-74976-9_32.
- [23] G. Ślusarczyk. A graph grammar approach to the design and validation of floor plans. *Computer-Aided Design*, 95:24–39, 2017. doi:10.1016/j.cad.2017.09.004.
- [24] G. Ślusarczyk, B. Strug, A. Paszyńska, E. Grabska, and W. Palacz. Semantic-driven graph transformations in floor plan design. *Computer-Aided Design*, 158:103480, 2023. doi:10.1016/j.cad.2023.103480.

Yufeng Liu received the Ph.D. degree from Hehai University, Nanjing, China. He is now an associate professor in College of Information Engineering, Nanjing University of Finance and Economics, China. His main research interests include software engineering, machine learning and visual language.



Yangchen Zhou received the bachelor degree from Nanjing University of Finance and Economics, China. He is now studying for a master's degree at College of Information Engineering, Nanjing University of Finance and Economics. His main research interests include graph grammar and generative design.

Fan Yang received the Ph.D. degree from Changchun University of Science and Technology, Changchun, China. He is now an associate professor in College of Information Engineering, Nanjing University of Finance and Economics, China. His main research interests include multimodal data fusion, machine learning, and deep learning.

Song Li received the Ph.D. degree from Anhui University, Hefei, China. He is now a lecturer in College of Information Engineering, Nanjing University of Finance and Economics, China. His main research interests include cloud security and applied cryptography.

Jun Wu received the Ph.D. degree from Nanjing University, Nanjing, China. He is now a lecturer in College of Information Engineering, Nanjing University of Finance and Economics, China. His main research interests include computational economics, algorithmic game theory, and mechanism design.

AN AGE-GROUP RANKING MODEL FOR FACIAL AGE ESTIMATION

Joseph D. Akinyemi^{1,*}  and Olufade F. W. Onifade² 

¹*Department of Computer Science, University of York, York, United Kingdom*

²*Department of Computer Science, University of Ibadan, Ibadan, Nigeria*

**Corresponding author: Joseph D. Akinyemi (akinyemijd@gmail.com)*

Abstract Age prediction has become an important Computer Vision task. Although this task requires the age of an individual to be predicted from a given face, research has shown that it is more intuitive and easier for humans to decide which of two individuals is older than to decide how old an individual is. This work follows this intuition to aid the age prediction of a face by exploiting the age information available from other faces. It goes further to explore the statistical relationships between facial features within age groups to compute age-group ranks for a given face. The resulting age-group rank is low-dimensional and age-discriminatory, thus improving age prediction accuracy when fed into an age predictor. Experiments on publicly available facial ageing datasets (FGnet, PAL, and Adience) reveal the effectiveness of the proposed age-group ranking model when used with traditional Machine learning algorithms as well as Deep Learning algorithms. Cross-dataset validation, a method of training and testing on entirely different datasets, was also employed to further investigate the effectiveness of this method.

Keywords: age estimation, age-group ranking, cross-dataset validation, dimensionality reduction, face processing, facial features.

1. Introduction

Ageing is a spontaneous and irreversible process of human life. This spontaneous and irreversible nature makes the ageing process non-linear and therefore difficult to predict. Thus, judging human age via facial appearance or other physical evaluations is difficult. Humans develop an innate ability, early in life to predict age to a reasonable degree of accuracy [18, 20], but this task still seems difficult for computers. The task of predicting or determining the age of an individual, given his/her facial image, is referred to in the Computer Vision and Image Processing research community as age estimation or age prediction. Automated age estimation has proven to have many interesting applications in security and surveillance, age-specific human-computer interaction, preventing age falsification, age-specific advertising etc. [2, 18].

Despite the success of deep learning for facial age estimation, the bulk of features are mostly learned directly from individual images without considering feature correlations across other images, especially with respect to the ages of those other images. This limits the relevance of learned features to the required discriminatory factor of ageing.

In this work, an age-group ranking approach is proposed, which exploits the relationships between faces across several age groups to enrich the extracted facial features for age estimation. The intuition behind this method is the observation that humans estimate ages by instinctively making comparisons between a given face with an unknown

age and other faces whose ages are known. This process is usually implicit and very fast with humans and it happens almost unconsciously. However, this process is influenced by the amount of exposure or experience of the person trying to estimate the age of another person. It could also involve scanning through faces in certain known age groups and trying to fix the questioned face in one of those age groups. Although it is difficult to completely model this process in a machine, we take intuition from this to develop an age group ranking model through which a questioned face is passed, compared with several age groups, and ranked accordingly. The resulting age-group rank is then used to embellish facial features to enhance the age-learning and prediction processes. The idea is to develop a model for extracting facial features that are age-discriminatory yet low-dimensional such that they can be used to predict ages from input face images. Experiments were performed on three publicly available facial ageing datasets FGnet [12], PAL [32] and Adience [17, 22] and a new dataset, FAGE, and the results obtained compete significantly with the state-of-the-art facial age estimation methods.

The specific contributions of this work include:

1. An age-group ranking model that produces age-discriminatory yet low-dimensional facial features from learned correlations between faces and age groups.
2. Deviation of Feature Values (DoFV) which allows age group ranks to be computed without requiring training or prior knowledge of the age of an input image.
3. An indigenous dataset (FAGE) of age-labelled facial images.
4. Cross-dataset validation to demonstrate the generalisation of the age-group ranking model.

The rest of the paper is organized as follows: Section 2 discusses related previous works in the field of facial age estimation, Section 4 discusses the methodology, Section 5 presents the experiments, results and discussion and Section 6 concludes the paper.

2. Related previous works

2.1. Using direct facial features for age estimation

One of the earliest works on facial age estimation was the work of Kwon and Lobo [24] which used face anthropometry and face wrinkles to describe the face and reported 100% accuracy on a set of 47 high-resolution face images classified as ‘Babies’, ‘Young Adults’ or ‘Seniors’. Research has since continued to produce several methods for improving facial age estimation using different face descriptors, different age representation methods, and various machine learning algorithms.

In [25], the Active Appearance Model (AAM) was used to represent the face and Principal Component Analysis (PCA) was used to obtain the deviation of each face from the mean AAM face model. In [19], an ageing pattern subspace learning model was proposed for facial age estimation. The authors defined an ageing pattern as a sequence

of personal face images sorted by time. Guo et al. [21] used Biologically Inspired Features (BIF) together with manifold learning techniques to estimate ages using Support Vector Machine (SVM) for age classification and Support Vector Regression (SVR) for age regression. Most of these methods, except for [19], directly used facial features of individuals for age classification or regression without considering possible relationships between faces with respect to age.

2.2. Using age ranking for age estimation

Some works have employed age ranking in various ways. In [8], the authors proposed a ranking approach to age estimation based on the intuition that humans estimate the age of an unknown individual by comparing his/her face to the faces of other individuals whose ages are known, thus resulting in a series of pairwise comparisons across a set of individuals with known ages. Based on this intuition, they proposed an age ranking model which results in binary classification-based comparisons. They used an ordinal ranking algorithm to reduce the ordinal ranking problem to a binary classification problem. [9] also proposed an age estimation algorithm that employed the relative order of ages as well as the classification costs. They maintained ordinal hyperplanes which separated all images into two groups based on the relative order of their age labels and used the cost of classification to find the best-separating hyperplane. In [3], an ethnic-specific age group ranking method was proposed for age estimation. In [7], age ranks were predicted based on a cost-sensitive hyperplane ranking algorithm, facial features were represented in low-dimensional space by a scattering transform so that exact ages are then predicted via category-wise age ranks. In [49], a deep learning model was used to rank faces and to estimate ages from faces. Ranking-CNN was proposed in [10] as a series of basic CNNs with binary outputs which were aggregated to obtain a final age label. Their experiments were conducted by pretraining their basic CNNs on Adience dataset [17] and then fine-tuning and validating it on the MORPH dataset with the best MAE of 2.96 years. While that work employed the ordinal age ranking between face pairs, ours employs ordinal relationships between each face and groups of faces in each age group.

2.3. Using deep learning for age estimation

More recently, deep learning models such as Convolutional Neural Networks (CNN) have been used to determine age from faces. [49] used a Scattering Network (a CNN variant) to develop a deep ranking model from age estimation. [35] used CNN with mean-variance and softmax losses to estimate ages from faces. [15] used CNN in a transfer learning setting to predict apparent as well as biological ages. [48] used CNN to learn the ordinal nature of ages for age estimation. In [47], a group- n age encoding was proposed, a CNN with multiple classifiers was used to learn the several age groups and a Local Age

Decoder was used to predict the exact ages. As accurate as deep learning models can be, they are computationally demanding and often require large amounts of training data.

3. Problem and motivation

Despite the impressive performance of many of these deep learning models, we observed that most of them failed to model the correlation of facial features with age groups as well as the inter-age groups' relationships. This is difficult for many of these models because deep learning architectures learn their features directly from inputs. Those which attempted to capture this relationship to an extent (e.g. [10, 47, 48]) still failed to capture the inter-age group relationships as it concerns facial features.

Also, most age ranking works conducted pairwise comparisons between faces leading to a large set of pairwise comparisons. Although DeepRank [49] does not rely on pairwise ranks, it infers its ranks from single images which still limits the possibility of capturing the correlation of faces within a larger set such as an age group. Secondly, most age-ranking works employed some form of learning to perform the age-ranking on faces. We also observed that in many cases, a reference image set was maintained for age ranking which is a subset of the training set and thus limits the amount of information available for age ranking. In [10], the age ranks were learned by several basic deep-learning networks, the results of which were aggregated to obtain a final age estimate. Considering the computational demand of deep networks, this could even be very expensive.

In this work, we propose an age-group ranking model which ranks face images by comparing an input image with every image in an entire training set and, in an attempt to represent age-group-specific features, derives an age group rank that is representative of each age group. Thus, each input image is ranked with respect to every image in a training set as well as with every age group in the training set. This provides a representation of the correlation of input images with every image in the training set as well as with every age group represented in the training set. Also, instead of learning and predicting age group ranks, we derived the deviation of feature values (DoFV) between compared faces and performed basic statistical computations on these values with respect to age groups, thus reducing the computational overhead that could have been incurred due to learning age ranks prior to learning exact ages.

4. Methodology

When a human is asked to estimate the age of a given facial image, several operations come into play in the mind. Apart from the fact that humans possess an innate ability to recognize age from the face, people generally tend to estimate age by comparing the given face to some other faces whose ages are known. This comparison is part of the innate ability and it is usually very fast and without prior thought or preparation. Thus,

a person's ability to correctly estimate age can be considerably impacted by his/her own age vis-a-vis his/her life experience [20, 38]. The more exposed and experienced a person is, the better is his/her age prediction ability. Thus, the age prediction ability of an adult is expected to be better than that of a child because of experience and the extent of development. In developing the proposed age-group ranking model, we leveraged this intuition.

Since a person's age estimation ability is impacted by his/her age and life experience, then the age ranking model can be enriched with more experience by providing more reference images for age ranking. Thus, our proposed age group ranking model employs its entire training image set in a leave-one-out fashion to rank images by their age groups. By using the leave-one-out method it is assured that no face image is ranked by comparison with itself. This is justifiable by the fact that the face whose age is in question should be compared with faces whose ages are known and not with itself, since its age is still unknown. Also, people within an age group tend to exhibit similar ageing features, thus making it easier to rank images by age groups than by exact ages. In fact, the sparse nature of ages in most facial ageing datasets makes it almost impossible to obtain enough images for each exact age rank. Also, unlike most other works, our age group ranking model does not learn age group ranks; rather, it obtains the deviation of feature values (DoFV) from compared faces and obtains the means and standard deviations of these deviation values within age groups which are then used to compute age group ranks for an input image.

However, there is still the challenge that, since the age of the face image in question is not known, it is difficult to decide which age group the image should be compared with in order to obtain an age group rank. To overcome this, the age group ranking model performs an exhaustive comparison of the questioned face with every face in every age group (in a dataset) so that the face is enriched with a representation of its correlation across various age groups. Consequently, the correlation of an input face with its actual age group is also learned from its comparison with several face images in that age group.

4.1. The age learning problem formulation

In this work, age estimation is modelled primarily as a regression problem. Thus, suppose we have a set A of face images and a set B of age labels ordered by the magnitude of the age values, the sets A and B can be represented as follows:

$$A = \{a_i | i = 1, \dots, p\}, \quad (1)$$

$$B = \{b_j | j = 0, \dots, q \wedge \forall j, b_{j+1} > b_j\}, \quad (2)$$

where a_i is face image, b_j is an age value, p is the number of face images and q is the highest age value. The expression $\forall j, b_{j+1} > b_j$ indicates that B is an ordered set, i.e., every age value is greater than the previous age value in the set, since age values are

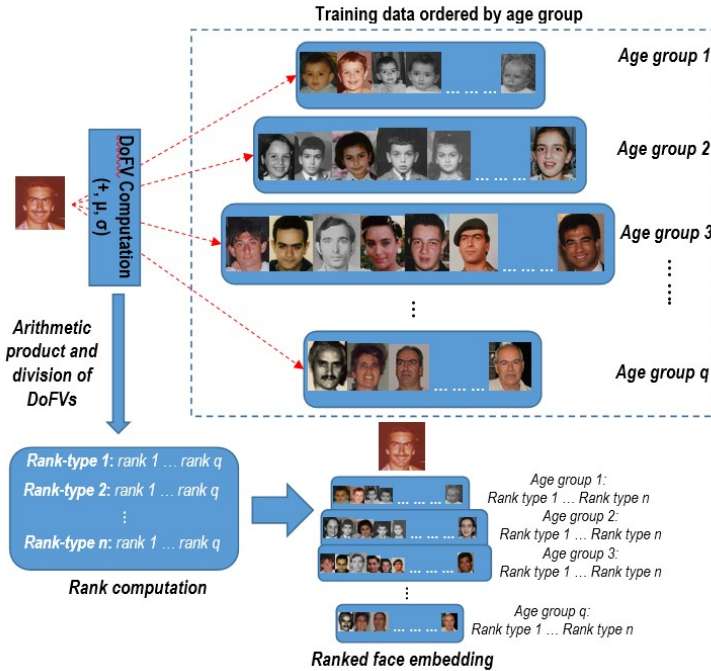


Fig. 1. The age-group ranking model.

ordered in time sequence. This ordering is necessary for age group ranking as we will see in subsection 4.2. Thus, the task of age estimation involves approximating an age learning function, say f_1 , which appropriately maps each facial image in A to its age value in B , according to

$$f_1(a_i) = b_j, \quad (3)$$

where $a_i \in A$ and $b_j \in B$.

4.2. The age-group ranking model

While age learning explores the relationship between face images and ages, age group ranking explores the relationships between each face image and other images in various age groups. Fig. 1 is a graphical illustration of how the AGR model ranks an input face by an age-group-ordered training set to derive different age group rank-types.

Following the definitions of the sets A and B above, we define a third set C of age

groups, according to

$$C = \{c_\lambda | \lambda = 1, \dots, w \wedge \forall \lambda, c_{\lambda+1} > c_\lambda\}, \tag{4}$$

where c_λ is an age group label and w is the number of age group labels.

Precisely, each $c_\lambda \in C$ is a subset of B . Thus each element of the set C of age groups is itself a set (of age values) contained in the set B and the sets c_λ are disjoint.

Further, the number of age groups in C is definitely less than the number of ages in B , that is $1 < w < q$.

The elements of each c_λ is determined from B by a range parameter, τ . Thus, we write $c_\lambda^\tau \subset B$.

Due to the nature of ageing and the challenge of insufficient data collection for its studies, the range parameter τ could be the same throughout the set C or may change for every $c_\lambda \in C$. This is necessary to ensure that the number of faces available to be mapped to each age group is relatively sizeable. However, as observed in (4), the ordering of B is retained in C as well. In our experiments, the value of τ was empirically determined based on the size of the dataset and the age distribution. This is necessary to ensure that the number of face images and their ages in each age group are sufficient for ranking a face, otherwise, we risk underrepresenting an age group.

Having defined the age learning function f_1 in (3), we further define an age group matching function h which maps faces to age groups, given the age of the face as follows:

$$h(a_i, b_j) = c_\lambda^\tau, \tag{5}$$

so that

$$\forall a_i \exists b_j, \text{ such that } f(a_i) = b_j, \tag{6}$$

and

$$\forall a_i \exists b_j, c_\lambda, \text{ such that } h(a_i, b_j) = c_\lambda^\tau. \tag{7}$$

While the age learning function has to be approximated (by training), the age group matching function simply associates a face (given its age) to its appropriate age group, thus it requires no approximation or training. However, the age group matching function only applies to training images or images whose ages are known and these are the images that make up the reference image set for comparison during age group ranking. As earlier stated, images to which an input image will be compared during age group ranking should be images whose ages or age groups are known, we, therefore, used all training images as the reference image set. The next challenge, however, is how to determine the age group to which an input (test) image belongs and this is where an age group ranking function steps in. It is noteworthy to state, therefore, that while the age group matching function simply assigns a face to an age group given the exact age of the face, the age group ranking function is responsible for capturing and representing the correlation of each

face with each age group. So, the age group matching function requires prior knowledge of the age of a given face so that it can construct the training set as a reference image set organized into age groups, but the age group ranking function requires no prior knowledge of the age of an input face.

Rather than approximating the age group ranking function by training, the function is realized by computing some arithmetic and statistical measures to represent the correlation of each face with each age group. Since the age group of the input (test) image is supposedly unknown, by collecting such measures for all age groups, we are able to capture the correlation of a face with various age groups. This further embellishes each face with relevant information for learning the discriminatory properties of faces in terms of their ages and age groups and reduces the overhead that could have been incurred by learning the age group ranks. The result of this operation is a multivariate age group rank for each face image representing its correlation with every age group.

Given the set A of face images and the set C of age groups as earlier defined, we define a tuple \vec{A} of sets of faces ordered by age groups as follows:

$$\vec{A} = (\hat{A}_1, \hat{A}_2, \dots, \hat{A}_w), \quad (8)$$

and

$$\hat{A}_\lambda = \{a_{\lambda_1}, a_{\lambda_2}, \dots, a_{\lambda_g}\}. \quad (9)$$

Each \hat{A}_λ , ($1 \leq \lambda \leq w$), is a set of face images matched to the age group c_λ , w is the number of age groups as indicated in equation (5), each a_{λ_j} , ($1 \leq j \leq g$) is a face image in the set \hat{A}_λ and g is the number of face images in a particular age group. Since \hat{A}_λ is a set, it means the face images in it are not necessarily ordered by age, but are definitely matched to the age group c_λ .

Given a face image a_i and a tuple \vec{A} of faces ordered by their age groups, the age group ranking function f_2 , which assigns an age group rank to image a_i to obtain an age-group-ranked face \hat{a}_i , is defined as follows:

$$f_2(a_i, \vec{A}) = \hat{a}_i. \quad (10)$$

At this point, each face image a_i has been transformed into a vector X_i of facial features; therefore, the age group rank \hat{r}_i of each face a_i is a vector obtained by computing the Deviation of Feature Values (DoFV) between each face and every face in the tuple \vec{A} of age grouped faces. The several operations abstracted in $f_2()$ are detailed in the following formulations.

Given a face a_i , with unknown age and age group, the age group rank \hat{r}_i of a_i is obtained as follows:

$$\varsigma(a_i, a_{\lambda_j}) = \Delta_{i\lambda_j}, \quad (11)$$

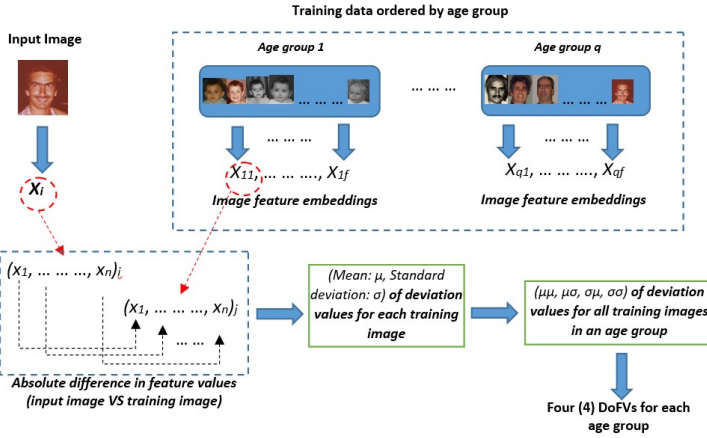


Fig. 2. DoFV computation.

where ς is the DoFV function, a_{λ_j} is the j th face in the set \hat{A}_λ of age-grouped faces and $\Delta_{i\lambda_j}$ is the obtained DoFV. DoFV is obtained by taking the absolute difference in feature values between an input image a_i whose age is unknown and an age grouped image a_{λ_j} whose age/age group is known. Then, for each age group, arithmetic and statistical measures of the differences in feature values are obtained for this particular input image and this provides the age group rank for the image at this particular age group. For each input image, this is repeated for all age groups and a vector of ranks is obtained for that input image, by concatenating the arithmetic and statistical measures of the DoFV obtained from all age groups. Therefore, the age group rank contains information about the statistical properties of images at feature, image, and age-group levels. Consequently, the age group rank obtained for each input image corresponds to the correlation of the feature values of the input image with the feature values of the various images in that age group. Hence, the obtained age group rank is actually a measure of the correlation of an input image with images of all age groups. With this information, the age learner (at training) can learn the correlation of each face with every age group, thus being able to better fit faces to their respective ages. Fig. 2 shows the DoFV computation procedure as explained above.

Suppose the facial features of a face image a_i is collected into the vector X_i of size n and each feature value in the vector X_i is indexed by t , ($1 \leq t \leq n$), then the following formulations can be stated for DoFV for a given face a_i as follows:

$$\Delta_t = |X_{it} - X_{\lambda_j t}|, \tag{12}$$

Δ_t being the DoFV for the t^{th} feature in the facial feature vector X_i , obtained as the

absolute difference between the t^{th} feature vector in the input face and the t^{th} feature vector in the j^{th} face of the age group \hat{A}_λ .

Then, for each face feature vector X_i ($1 \leq i \leq p$; p being the number of face images), two arithmetic and statistical measures of the DoFV are taken, namely the arithmetic mean and the standard deviation denoted as $\Delta_{i\lambda_j}^\mu$ and $\Delta_{i\lambda_j}^\sigma$, respectively.

Subsequently, for each age group, four arithmetic and statistical measures are obtained as mean of means ($\Delta_{i\lambda}^{\mu\mu}$), mean of standard deviations ($\Delta_{i\lambda}^{\mu\sigma}$), standard deviation of means ($\Delta_{i\lambda}^{\sigma\mu}$) and standard deviation of standard deviations ($\Delta_{i\lambda}^{\sigma\sigma}$), as shown in equations (13) to (16), respectively.

$$\Delta_{i\lambda}^{\mu\mu} = \frac{\sum_{j=1}^g \Delta_{i\lambda_j}^\mu}{g} \quad (13)$$

$$\Delta_{i\lambda}^{\mu\sigma} = \frac{\sum_{j=1}^g \Delta_{i\lambda_j}^\sigma}{g} \quad (14)$$

$$\Delta_{i\lambda}^{\sigma\mu} = \sqrt{\frac{\sum_{j=1}^g (\Delta_{i\lambda_j}^\mu - \Delta_{i\lambda}^{\mu+})^2}{g-1}} \quad (15)$$

$$\Delta_{i\lambda}^{\sigma\sigma} = \sqrt{\frac{\sum_{j=1}^g (\Delta_{i\lambda_j}^\sigma - \Delta_{i\lambda}^{\sigma+})^2}{g-1}} \quad (16)$$

For every face image a_i , these four values are obtained for each age group resulting in $4 \times w$ values (w being the number of age groups), since the age/age group of the query face is supposedly unknown.

The age group rank \hat{r}_i is obtained by performing arithmetic multiplication and division operations between these four values in eight different ways. These eight values are computed for each age group, giving a maximum of $8 \times w$ (w being the number of age groups) values making up the age group rank of each image. The selected eight values, called rank-types, are computed as $\varpi_{i\lambda_1} = \Delta_{i\lambda}^{\mu\mu} \times \Delta_{i\lambda}^{\sigma\mu}$; $\varpi_{i\lambda_2} = \Delta_{i\lambda}^{\mu\sigma} \times \Delta_{i\lambda}^{\sigma\sigma}$; $\varpi_{i\lambda_3} = \Delta_{i\lambda}^{\mu\mu} / \Delta_{i\lambda}^{\sigma\mu}$; $\varpi_{i\lambda_4} = \Delta_{i\lambda}^{\mu\sigma} / \Delta_{i\lambda}^{\sigma\sigma}$; $\varpi_{i\lambda_5} = \Delta_{i\lambda}^{\mu\mu} \times \Delta_{i\lambda}^{\mu\sigma}$; $\varpi_{i\lambda_6} = \Delta_{i\lambda}^{\sigma\mu} \times \Delta_{i\lambda}^{\sigma\sigma}$; $\varpi_{i\lambda_7} = \Delta_{i\lambda}^{\mu\mu} / \Delta_{i\lambda}^{\mu\sigma}$ and $\varpi_{i\lambda_8} = \Delta_{i\lambda}^{\sigma\mu} / \Delta_{i\lambda}^{\sigma\sigma}$, where $\varpi_{i\lambda_1}, \varpi_{i\lambda_2}, \dots, \varpi_{i\lambda_8}$ are the eight rank-types. For space constraints, we leave out the equations for these ranks as they can be easily deduced from equations (13)-(16).

Consequently, the rank \hat{r}_i ($1 \leq i \leq p$; p being the number of face images) of each image is made up by concatenating the obtained rank values of all the age groups for

each rank type, as follows:

$$\tilde{r}_{ik} = \varpi_{i1_k} \oplus \varpi_{i2_k} \oplus \dots \oplus \varpi_{iw_k} , \tag{17}$$

where $\varpi_{i1_k}, \varpi_{i2_k}, \dots, \varpi_{iw_k}$ are the values for rank-type k ($1 \leq k \leq 8$) for each of the w age groups and \tilde{r}_{ik} is the resulting vector for rank-type k for all age groups. Finally, the rank \hat{r}_i of an image a_i for all rank types is given as

$$\hat{r}_i = \tilde{r}_{i1} \oplus \tilde{r}_{i2} \oplus \dots \oplus \tilde{r}_{it} , \tag{18}$$

where t is the number of different rank-types and in this case, $t = 8$. Eventually, the age group rank obtained for a face image a_i is concatenated with the facial features of a_i to obtain an age-group-ranked face image \hat{a}_i as stated in equation (17). Thus, we can write

$$\hat{X}_i = X_i \oplus \hat{r}_i , \tag{19}$$

where \hat{X}_i is the age-group-ranked feature vector of the age-group-ranked face \hat{a}_i . Equation (3) can therefore be rewritten as in equation (20) so that a learning algorithm can then approximate this function:

$$f_1(\hat{X}_i) = b_j . \tag{20}$$

The effect of this is that the learning algorithm has more age-relevant facial features to learn from in approximating this function and thereby estimating the exact age of a given face. Details of the learning algorithms are given in the next section.

Summarily, the entire process described produces enhanced features (low-dimensional and discriminatory) that can be supplied as input to a learning algorithm to predict the exact age of a given face. Links to the dataset and source code will be made available after acceptance.

5. Experiments, Results, and Discussions

5.1. Experimental Settings

Our age group ranking (AGR) model was implemented in MATLAB R2016a. We used Local Binary Patterns (LBP) [34], raw image pixel features and deep features (VGG16 [45], Inception-V3 [46], Xception [11] and VGGFace [36]) as face descriptors and used Support Vector Regression (SVR) with Radial Basis Function (RBF) kernel (to capture the non-linearity of face ageing) for age learning. Experiments were performed on four different facial ageing datasets, namely FGnet [12], which contains 1002 images of 82 individuals, PAL [32], with 1046 images of 575 individuals and a new dataset, FAGE (Facial expression, Age, Gender and Ethnicity) with 540 images of 328 individuals, and Adience [17]. For Adience dataset, the age labels are not exact ages but age groups, therefore in place of SVR, we used the Discriminant Analysis classifier with a

quadratic kernel, henceforth referred to as Quadratic Discriminant Analysis (QDA), for age group learning. For SVR, the age learning optimization algorithm used was Sequential Minimal Optimization. The estimated Lagrange multipliers for the support vectors as well as the optimization coefficients were initialized to zero and training was done for 1000 iterations. For QDA, the misclassification cost was a square matrix whose values were derived from the distance between the age classes and the prior probabilities were empirically determined from the frequencies of the age classes.

Although our model was originally formulated for regression, in the case of Adience dataset, the model is adapted to classification by using the supplied age groups both for age group ranking and as the responses to be learned in age classification, so Adience does not require the age group matching function of equation (5). As will be seen in Tab. 1, the age groups in Adience are already too wide and too few (only eight of them), so merging two or three age groups into one will only increase the age gap and reduce the number of age groups available for age group ranking. As will be seen in the results, this limitation affected the result of age group ranking on Adience dataset.

Our choice of these datasets is because they are publicly available and have long-standing usage in age estimation research. FAGE was collected for this research, specifically to investigate age estimation on indigenous African faces (a problem rarely studied). To investigate the generalization ability of the trained models, we also performed cross-dataset validation (which is rarely done because of the peculiarities of each dataset) between three of the four datasets studied (Adience was excluded as it does not include exact ages).

For training and validation on FGnet, we adopted the popular subject-exclusive Leave-One-Person-Out (LOPO) cross-validation protocol as described in [19]. For PAL and FAGE datasets, we used 5-fold cross-validation and for Adience, we used the subject-exclusive 5-fold cross-validation as suggested in [17]. The evaluation metrics that have become standards for age estimation are Mean Absolute Error (MAE) and Cumulative Score (CS). MAE is the average of the absolute difference between the actual and predicted ages while CS is the percentage of the dataset whose ages are correctly predicted at a given error level. However, for Adience, the recommended and popular evaluation metric is the percentage classification accuracy (ACC) and is usually divided into exact accuracy and 1-off accuracy (taking as correct, predictions off by one age group). Thus, with MAE, the lower the value, the better the performance, while with ACC and CS, the higher the value, the better the performance.

Each dataset was split into age groups such that each age group spanned about five years (i. e. $\tau \approx 5$) except in cases where there were not enough images to represent an age group. For Adience, we simply used the age group classes that came with the dataset as the age groups for ranking. Tab. 1 shows the division of the age groups within each of the four datasets. Age group ranking was thus performed on each dataset using these age group divisions, thus resulting in 11, 12, 10, and 8 age group ranks for FGnet, PAL,

Tab. 1. Datasets divisions by age group.

Adience Dataset		FAGE Dataset		FGnet Dataset		PAL Dataset	
Age group	# faces	Age group	# faces	Age group	# faces	Age group	# faces
0 – 2	2509	0 – 5	44	0 – 4	194	18 – 20	116
4 – 6	2140	6 – 10	97	5 – 8	153	21 – 25	274
8 – 13	2292	11 – 15	66	9 – 12	135	26 – 30	86
15 – 23	1887	16 – y20	71	13 – 16	130	31 – 35	44
25 – 36	5549	21 – 25	142	17 – 20	118	36 – 40	34
38 – 46	2429	26 – 30	63	21 – 24	64	41 – 45	38
48 – 58	937	31 – 35	27	25 – 28	51	46 – 50	34
60 – 100	872	36 – 40	10	29 – 32	38	51 – 55	40
–	–	41 – 45	13	33 – 36	36	56 – 60	12
–	–	46 – 80	7	37 – 40	23	61 – 70	162
–	–	–	–	41 – 69	60	71 – 80	139
–	–	–	–	–	–	81 – 93	67
Total	18615	Total	540	Total	1002	Total	1046

FAGE, and Adience datasets, respectively. For brevity, AGR refers to age group ranking in all tables and figures where it appears.

A note on Adience dataset

According to [17], the Adience dataset is said to contain 26 580 images of 2 284 subjects. However, the dataset downloadable from the authors' website contains exactly 19 370 images (see Table I of [37]) out of which only 18 615 images are labelled with age groups. This is further confirmed by our observation of the fact that the breakdown provided in Table II in [17] does not in any way add up to 26 580 images. More so, we observed that the age labels in the available dataset (from their website) are somewhat inconsistent with what is provided in the paper. We worked around this to aggregate the scattered pieces of age labels into coarse age groups and we eventually ended up with eight labels similar to the ones indicated in [17], but some of our age groups covered wider ranges.

Face preprocessing and feature extraction

Each face image was preprocessed by converting it into an 8-bit grayscale image (if coloured) resulting in pixel intensity values between 0 and 255. From the grayscale image, the face was detected and aligned using a multi-stage method described in [4]. Before feature extraction, images were resized to various sizes depending on the feature descriptor to be used. For LBP and raw image pixels features, images were resized to 120×100 pixels; for VGG16 and VGGFace features, images were resized to 224×224 pixels; for Inception-V3 and Xception, images were resized to 299×299 pixels. For raw pixels and LBP features, feature histograms were obtained from ten (10) face regions

defined around the forehead, the outer eye corners, the inner eye corners, the area under the eyes, the area between the two eyes, the nose bridge, the nose lines, the cheek area, the cheekbone areas, and the periocular face region. Features histograms from each defined face region were aggregated and compacted using the method in [5]. We selected compaction ranges of 5 and 10 for raw pixels and LBP, respectively. For LBP features, $LBP_{8,1}$ (8-pixel neighbourhood and pixel distance/radius of 1) was used. The resulting features from each descriptor were then used to rank each face as described in the previous section and to obtain age group ranks for each face for all age groups. The resulting age group ranks were passed into SVR/QDA for age/age-group learning and prediction. We then carried out comparative analyses of the performance of age group ranking on each dataset and each feature descriptor.

5.2. Dataset-specific results

To investigate the impact of our AGR model, we trained SVR/QDA on:

1. the entire features vector before age group ranking (high-dimensional features);
2. the entire features along with the age group ranks (high-dimensional features);
3. the age group ranks alone (low-dimensional features).

Each feature type (before and after age group ranking), was normalized by scaling the feature values to a narrow interval $(0, 1)$ using the standard deviation and means of the feature values. The MAEs obtained in each case are reported in Tab. 2. The value of x in Tab. 2 refers to the number of rank-types multiplied by the number of age groups in each dataset. So, from Tab. 1 and Tab. 2, it can be inferred that $x = 64, 80, 88,$ and 96 for Adience, FAGE, FGnet, and PAL datasets respectively. From Tab. 2, it is obvious that the age group ranks significantly reduced the age estimation error in all cases even though it provides significantly low-dimensional features for age learning.

We further investigated the performance of each of the eight (8) rank-types for age estimation and reported the results in Tab. 3. From Tab. 3, it can be observed that rank-types 3, 4, and 6 generally gave the lowest MAE (values in boldface). For all raw pixel features, rank-types 4 and 6 seem to give the best performance, except on PAL dataset where rank-type 8 performed better than the two and that was the only instance where rank-type 8 performed the best in the entire experiment. For LBP features, rank-types 3 and 6 gave the best performances. For both VGG16 and VGGFace features, rank-types 4 and 6 were the best. For Inception and Xception features, rank-types 3 and 6 were the best; in fact, with Xception, rank-type 3 consistently outperformed rank-type 6 on all datasets. On Adience dataset, the best performing rank-types are rank-types 3 and 6; on FAGE dataset, the best performing is rank-type 6; on FGnet dataset, the best performing are rank-types 3, 4, and 6, but predominantly 4; while on PAL dataset, the best performing are rank-types 3, 4, 6 and 8 (but the good performance of rank-type 8 is more like an outlier in the entire set of experiments).

Tab. 2. MAE of age estimation results before and after age group ranking. *Ftr.* stands for *feature(s)* and *dim.* stands for *dimensionality*.

Experiment setting	Ftr. type	Ftr. dim.	ACC [%]	MAE (years)		
			Adience	FAGE	FGnet	PAL
Before AGR (features only)	Raw pixel	520	(31.30, 56.79)	7.02	8.43	14.44
	LBP	260	(29.59, 58.09)	6.56	8.36	12.32
	VGG-16	4096	(19.06, 52.12)	6.25	6.94	10.39
	VGGFace	2622	(18.89, 43.31)	5.18	4.65	5.07
	Incep-V3	2048	(22.67, 41.97)	6.49	6.14	12.34
	Xception	2048	(19.82, 36.51)	6.97	6.78	11.96
After AGR (features + ranks)	Raw pixel	520+ x	(36.93, 59.93)	6.72	8.36	13.23
	LBP	260+ x	(43.83, 64.54)	4.29	4.99	7.29
	VGG-16	4096+ x	(19.25, 52.18)	6.10	6.77	10.19
	VGGFace	2622+ x	(18.71, 42.52)	5.05	4.52	5.00
	Incep-V3	2048+ x	(25.06, 45.72)	6.44	5.83	12.05
	Xception	2048+ x	(17.93, 33.83)	6.95	6.26	11.68
After AGR (ranks only)	Raw pixel	x	(60.24, 71.70)	6.22	7.27	12.44
	LBP	x	(61.75, 75.48)	3.11	2.98	5.17
	VGG-16	x	(53.02, 74.94)	3.55	3.51	6.36
	VGGFace	x	(67.90, 90.28)	3.71	2.84	4.52
	Incep-V3	x	(52.47, 75.38)	6.70	3.25	13.37
	Xception	x	(52.68, 75.26)	6.88	3.43	11.60

This is significant as it shows that we can even lower age estimation error by using just one of the rank-types, thereby dropping the dimension of features needed for age learning from x to $x/8$; meaning just 8 feature dimension for Adience, 10 for FAGE, 11 for FGnet and 12 for PAL datasets. One observable similarity in the computation of these three best-performing rank-types is the fact that they all involve either the standard deviation of means ($\sigma\mu$) or the mean of standard deviations ($\mu\sigma$) as described in Subsection 4.2. This shows that the combination of statistical and arithmetic measures of the facial features properly captured the relationship between facial features within and across age groups in low dimensions.

As expected, the performance of these rank-types on Adience is still relatively poor. This is due to the few age groups *vis-a-vis* the dataset size – there are only 8 age groups for ranking over 18000 images. For this reason, we investigated the combination of the different best-performing rank-types as well as the best-performing feature types on Adience and reported the results in Tab. 4. Interestingly, with the proper combinations of rank-types as well as feature types, the performance improves significantly and the best result was obtained with the combination of rank-types 3 and 6 on the combination

Tab. 3. MAE of age estimation with each rank-type (rt). Only exact ACC is shown for Adience.

Ftr. type rt 1 to 8	ACC [%]	MAE (years)		
	Adience	FAGE	FGnet	PAL
Raw pixel	48.79, 52.64, 53.58, 55.63, 28.69, 62.35 , 26.56, 33.45	5.43, 6.17, 4.52, 4.71, 6.11, 4.23 , 7.15, 6.64	6.68, 7.05, 5.51, 4.93 , 7.42, 5.02, 9.25, 8.25	13.61, 15.07, 13.45, 13.35, 10.44, 11.77, 13.3, 9.84
LBP	51.34, 30.69, 57.05 , 36.47, 28.95, 49.70, 24.58, 34.77	1.93, 2.98, 2.40, 3.18, 4.74, 1.71 , 7.21, 4.88	1.88, 3.35, 2.17, 3.27, 7.63, 1.79 , 9.61, 5.44	4.21, 7.92, 3.53, 7.05, 10.14, 3.29 , 12.99, 8.09
VGG16	38.13, 43.11, 45.72, 48.77 , 36.43, 46.38, 35.64, 38.80	3.13, 2.21, 2.91, 2.57, 5.02, 2.10 , 6.78, 4.45	5.32, 3.71, 3.30, 2.55 , 5.61, 2.67, 6.92, 5.85	9.57, 7.22, 5.72, 5.10 , 7.27, 5.71, 11.11, 8.49
VGGFace	52.48, 54.41, 66.99, 67.82 , 58.04, 63.22, 30.08, 31.82	3.22, 2.95, 3.14, 3.01, 4.36, 2.20 , 7.10, 7.09	3.77, 3.41, 2.05, 1.96 , 3.74, 2.09, 7.67, 6.86	6.27, 5.87, 4.27, 3.99 , 4.47, 4.40, 11.84, 13.66
Incep-V3	42.30, 47.17, 47.02, 45.12, 39.04, 49.65 , 35.41, 38.49	5.76, 6.42, 5.25, 5.48, 5.97, 5.04 , 7.48, 6.36	4.29, 4.27, 2.44 , 2.61, 5.52, 2.76, 7.74, 6.84	14.82, 14.30, 11.03 , 11.34, 12.93, 11.73, 15.23, 14.26
Xception	36.41, 45.20, 46.62 , 45.14, 39.33, 46.61, 36.36, 39.37	7.20, 6.95, 5.96 , 5.95, 6.62, 5.99, 7.10, 7.00	5.02, 4.31, 2.84 , 2.84 , 5.56, 3.37, 7.80, 6.17	12.44, 11.95, 9.72 , 9.97, 10.65, 9.93, 14.03, 13.53

of VGGFace, LBP, Raw Pixel, Inception, and Xception features. Fig. 3 shows sample images from the four datasets for which age prediction with AGR succeeded and those for which it failed using the best-performing features.

Tab. 5 shows some of the most recently reported state-of-the-art results on Adience, FGnet, and PAL datasets (FAGE is a relatively new dataset, so there are no existing methods on it to compare with). In the table, the asterisk (*) in the third column (ftrs. dim.) refers to those in which the exact feature dimension was not explicitly reported in the literature. However, it is common knowledge that most of the deep learning features are in the order of thousands, while our method uses features in the order of tens. From Tab. 5, it is seen that our method competes significantly with the best of these methods achieving the lowest MAEs on FGnet (1.79 years) and PAL (3.29 years) and the best exact accuracy (85.1%) on Adience; VLRIX stands for the combination of VGGFace, LBP, Raw pixel, Inception and Xception features as seen in the third to the last row of Tab. 4. We consider this a significant achievement considering the highly reduced feature dimension generated by our AGR model and the fact that it achieves this even with fairly simple feature extraction techniques (raw pixel and LBP), thus making our

Tab. 4. Different combinations of rank-types and feature types on Adience dataset. Abbreviations: I – Inception, L – LBP, R – Raw pixel, V – VGGFace, V16 – VGG16, X – Xception.

Rank-types	Feature types	Ftr. dim.	ACC (%)	
			Exact \pm std.	1-off \pm std.
3, 4	All	96	83.7 \pm 2.10	93.2 \pm 1.07
3, 6	All	96	84.0 \pm 2.79	93.9 \pm 1.26
4, 6	All	96	82.1 \pm 2.91	93.1 \pm 1.54
3, 4, 6	All	144	83.7 \pm 2.56	93.6 \pm 1.22
3, 4	X, I	32	55.8 \pm 4.31	78.4 \pm 2.23
3, 6	X, I, L, R	64	79.4 \pm 2.12	89.5 \pm 1.18
4, 6	V16, V, L, R, I	80	83.2 \pm 2.88	93.4 \pm 1.34
4, 6	V16, V, L, R, X	80	83.4 \pm 3.02	93.6 \pm 1.47
3, 6	V16, V, L, R, X	80	84.8 \pm 3.11	94.2 \pm 1.21
3, 6	V16, V, L, R, I	80	84.5 \pm 2.88	94.0 \pm 1.38
3, 6	V, L, R, I, X	80	85.1 \pm 2.33	94.6 \pm 0.88
3, 4, 6	V, L, R	72	85.5 \pm 3.12	94.3 \pm 1.15
3, 4, 6	V, L, R, V16	96	84.6 \pm 2.99	93.7 \pm 1.39

results more easily reproducible. All these results had been achieved with features of relatively low dimension – 80 on Adience, 11 on FGnet, and 12 on PAL.

CS often gives a better picture of the performance of an age estimation algorithm at different levels of the prediction error. We plotted our CS scores along with some of the best results on FGnet for which CS plots were reported and compared the results. Fig. 4 further confirms the significant improvement offered by our AGR model (AGR-LBP-r6 and AGR-VGGFace-r4) on FGnet. At an error level of 0, only EBIF [14] started ahead of the AGR model and AGR overtook it at error level 1. AGR performs at par with GEF up to error level 1 after which AGR significantly overtakes. Generally, from error level 2 upwards, AGR outperforms all the compared methods and finishes far ahead of them with CS of 95% at error level 5 and 99% at error level 10. Previous works on PAL rarely report their CS scores so there will be no basis for such comparisons, thus we leave out the CS curve on PAL. Also, because the FAGE dataset is new, there are no previous results with which we can compare it.

5.3. Cross-Dataset Validation

To better study the generalization of our model, we performed cross-dataset validation in two settings:

1. on FGnet and PAL datasets;
2. on FGnet, PAL, and FAGE datasets.

Tab. 5. Comparison with previous results on Adience, FGnet and PAL. rt: rank-type. Note the 3rd column: filters dimension.

Method	Year	filters dimension (Adience, FGnet, PAL)	ACC [%]	MAE (years)	
			Adience (Exact, 1-off)	FGnet	PAL
EBIF [14]	2011	EBIF*	–	3.17	–
W-RS [50]	2013	100–900	–	–	5.99
Joint-Learn [6]	2014	LBP*	–	–	5.26
DeepRank [49]	2015	500	–	–	4.31
GEF [30]	2015	LBP,BIF,HOG*	–	2.81	–
CNN [26]	2015	CNN ftrs.*	(50.7, 84.7)	–	–
DA [39]	2017	VGG-16 ftrs.*	(60.0, 94.5)	–	–
DNN [41]	2017	VGG-16 ftrs.*	(62.8, 95.8)	–	–
ODFL [28]	2017	CNN ftrs.*	–	3.89	–
All-in-one [37]	2017	CNN ftrs.*	–	2.00	–
DEX [40]	2018	VGG-16 ftrs.*	(64.0, 96.6)	3.09	–
Group-n [47]	2018	VGG-16 ftrs.*	–	2.96	–
DRF [42]	2018	VGG-16 ftrs.*	–	3.85	–
CNN2ELM [16]	2018	CNN ftrs.*	(66.49, –)	–	–
Joint-Learn [31]	2018	$LBP_{(8,1)}$	–	–	5.26
MVL [35]	2018	CNN ftrs.*	–	2.68	–
BridgeNet [27]	2019	CNN ftrs.*	–	2.56	–
TransLearn [15]	2019	4096 VGG-16 ftrs.	–	–	3.79
SORD [13]	2019	VGG-16 ftrs.*	(59.6, –)	–	–
ODL [29]	2019	VGGFace ftrs.*	–	2.92	3.99
DDRF [43]	2019	VGG-16 ftrs.*	–	3.47	–
C3AE [51]	2019	*	–	2.95	–
DOEL [48]	2020	ResNet ftrs. *	–	3.44	–
DLC [1]	2020	CNN ftrs.*	(83.1, 93.8)	–	–
SR [33]	2020	CNN ftrs.*	–	–	8.33
DCN [23]	2022	VGG ftrs.*	–	2.13	–
ABC+Swin [44]	2023	Transformer ftrs.*	(56.1, –)	2.52	–
AGR-LBP (rt6)	Ours	[8, 11, 12]	(49.7, 68.9)	1.79	3.29
AGR-VLRX (rt3+rt6)	Ours	[80, –, –]	(85.1, 94.6)	–	–

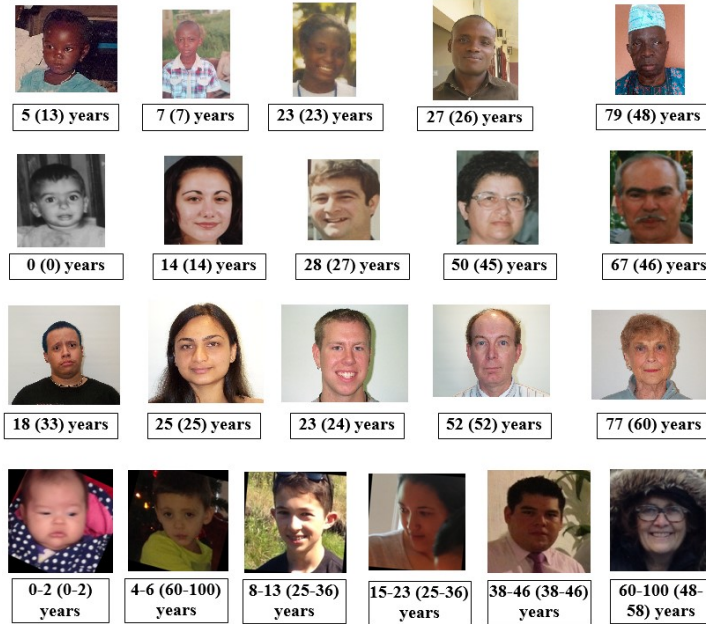


Fig. 3. Sample images and their true/predicted ages. from the 1st to the last row: FAGE, FGnet, PAL and Adience. Predicted ages are in parentheses.

In both settings, we used LBP (rank-type 6) and VGGFace (rank-type 4) features since they were the two best-performing features. In the second setting, we trained and tested the model on a combination of FGnet, PAL, and FAGE datasets. The Adience dataset is not used for Cross-dataset validation because it does not contain exact ages and is therefore unsuitable for a regression task as is the case with the other 3 datasets.

In setting 1, since both datasets cover separate age ranges, we selected the intersection of the age ranges covered (i. e. 18-69 years) and selected all faces falling within this age range. We found 362 FGnet images and 820 PAL images within this age range, making 1182 images altogether. We then ranked this new set of 1182 images on the entire set of FGnet and referred to it as FG-ranked, we also ranked it on the entire set of PAL images and referred to it as PAL-ranked. We trained and tested FG-ranked and PAL-ranked datasets using 5-fold cross-validation and obtained MAEs of 8.86 and 6.27 years with LBP features and 4.55 and 4.32 years with VGGFace features on FG-ranked and PAL-ranked datasets, respectively. As expected, the MAEs are higher in the cross-dataset environment, however, the result is worse when FGnet images are used to rank the data. This is because FGnet has 44 images less than PAL and FGnet contains 7 missing ages,

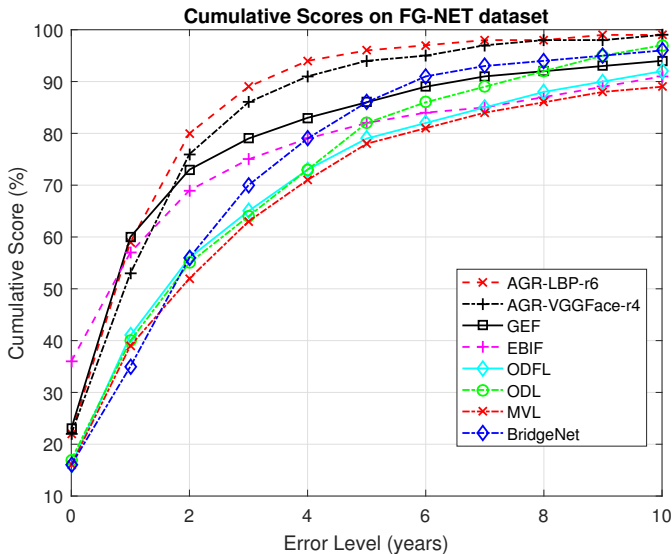


Fig. 4. CS curves of best-reported works on FGnet

while PAL contains only 1 missing age. PAL also covers a wider age range and contains more images for its age groups than FGnet. This goes to show that with more images available for age ranking and more ages represented within each age group, AGR offers better performance.

In the second setting, because of the differences in the number of age groups in each of the combined datasets, we created a new set of 15 age groups covering all the age groups in all three datasets and ranked each image in the combined dataset on this. There are a total of 2715 images in the combined dataset. We trained and tested with 5-fold cross-validation and obtained MAEs of 4.03 years and 4.33 years for VGGFace and LBP, respectively. However, the increased error rate is attributed to the ethnic diversity of the three datasets and the possibility that the age groups have become relatively too much for the dataset size.

The improved performance of VGGFace over LBP is an indication of the expressiveness of deep features in more complicated settings such as cross-dataset validation and with more data (as in setting 2). Generally speaking, the MAEs in both cross-dataset validation settings did not soar beyond expectations despite the wide inter-dataset variations; this is a pointer to the robustness of the AGR model and the intuition of age group ranking.

6. Conclusion

In this work, an age group ranking approach for facial age estimation was developed. The developed model uses the intuition that age can be better estimated from faces when there is sufficient information about other faces in several different age groups to rank a query face. The developed method was tested and validated on four datasets (FAGE, FGnet, PAL, and Adience). Experiments were performed on these datasets using standard protocols and the results compete significantly with the state-of-the-art age estimation methods. We further investigated the generalization of the method using cross-dataset validation and it turned out that the developed AGR method gives relatively good performance even across different datasets. The intuition of age group ranking developed here is superior to the existing age ranking methods in that age group ranking ranks images by age group rather than by exact ages thus making more data available for an image to be ranked. This is done without the need for prior knowledge of a particular age group rank via learning as the age ranking model uses available aging information from all age groups to rank a given face. More interestingly, the AGR model does not depend extensively on deep learning models as in current works but still competes significantly with deep-learning-based age estimation models. The findings from this work show that despite the impressive results of deep learning in recent times, the impact of age group ranking on face-based age estimation is indeed significant and should not be discarded. This work has also shown that age estimation via age-group ranking is more intuitive and gives better performance than direct age estimation from a single face.

The major limitation of the AGR model is that it does not fit directly into a deep learning architecture as it requires features to be extracted and enhanced before it is being passed to a classifier/regressor. However, the AGR model works when on simple features such as raw pixels as well as deep features as the features are further enriched with age group information before they are passed into a classifier/regressor.

Future works could consider building deep learning models that can explore the relationship between faces in terms of their age groups while estimating the age of a given face image. Future works could also consider using more rank-types and different age groupings to understand the impact of the number of age groups *vis-a-vis* the age range and the number of images within each age group. Considering the impact of the statistical measures of variation used in DoFV, there is a need to explore more statistical measures that could improve age estimation accuracy.

Acknowledgments

The authors would like to thank Prof. Andreas Lanitis for providing his copy of the FGnet dataset for this research. We also like to thank Dr. E. J. Dansu for his contribution in reviewing and correcting the mathematical equations.

References

- [1] O. Agbo-Ajala and S. Viriri. Deeply learned classifiers for age and gender predictions of unfiltered faces. *Scientific World Journal*, 2020:1289408, 2020. doi:10.1155/2020/1289408.
- [2] J. D. Akinyemi. *GWAgeER; A GroupWise Age-Ranking Approach to Age Estimation from Still Facial Image*. Master's thesis, University of Ibadan, Ibadan, 2014. 161 pages. doi:10.13140/RG.2.1.2495.1763, <https://ibadan.academia.edu/AkinyemiDamilola/Theses>.
- [3] J. D. Akinyemi and O. F. W. Onifade. An ethnic-specific age group ranking approach to facial age estimation using raw pixel features. In: *Proc. 2016 IEEE Symposium on Technologies for Homeland Security (HST)*, pp. 1–6. IEEE, Waltham, MA, USA, 10–11 May 2016. doi:10.1109/THS.2016.7819737.
- [4] J. D. Akinyemi and O. F. W. Onifade. A computational face alignment method for improved facial age estimation. In: *Proc. 2019 15th International Conference on Electronics, Computer and Computation (ICECCO)*, pp. 1–6. IEEE, Abuja, Nigeria, 12 2019. doi:10.1109/ICECCO48375.2019.9043246.
- [5] J. D. Akinyemi and O. F. W. Onifade. Facial age estimation using compact facial features. In: *Computer Vision and Graphics: Proc. International Conference on Computer Vision and Graphics (ICCVG) 2020*, vol. 12334 of *Lecture Notes in Computer Science*, pp. 1–12. Springer International Publishing, Warsaw, Poland, Sep 14–16 2020. doi:10.1007/978-3-030-59006-2_1.
- [6] F. Alnarjar and J. Alvarez. Expression-invariant age estimation. In: *Proc. 25th British Machine Vision Conference (BMVC) 2014*, pp. 28.1–28.11. Nottingham, UK, 1–5 Sep 2014. (doi inoperative). doi:10.5244/C.28.14, <https://bmva-archive.org.uk/bmvc/2014/papers/paper081/index.html>.
- [7] K.-Y. Chang and C.-S. Chen. A learning framework for age rank estimation based on face images with scattering transform. *IEEE Transactions on Image Processing*, 24(3):785–798, 2015. doi:10.1109/TIP.2014.2387379.
- [8] K.-Y. Chang, C.-S. Chen, and Y.-P. Hung. A ranking approach for human ages estimation based on face images. In: *Proc. 2010 20th International Conference on Pattern Recognition (ICPR)*, pp. 3396–3399. Istanbul, Turkey, 23–26 Aug 2010. doi:10.1109/ICPR.2010.829.
- [9] K.-Y. Chang, C.-S. Chen, and Y. P. Hung. Ordinal hyperplanes ranker with cost sensitivities for age estimation. In: *Proc. 2011 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 585–592. Colorado Springs, CO, USA, 20–25 Jun 2011. doi:10.1109/CVPR.2011.5995437.
- [10] S. Chen, C. Zhang, M. Dong, J. Le, and M. Rao. Using Ranking-CNN for age estimation. In: *Proc. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 742–751. Honolulu, HI, USA, 21–26 Jul 2017. doi:10.1109/CVPR.2017.86.
- [11] F. Chollet. Xception: Deep learning with depthwise separable convolutions. In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1800–1807. Honolulu, HI, USA, 21–26 Jul 2017. doi:10.1109/CVPR.2017.195.
- [12] T. F. Cootes, G. Rigoll, E. Granum, J. L. Crowley, S. Marcel, et al. Face and Gesture Recognition Working group, 2002. <http://www-prima.inrialpes.fr/FGnet/>, FGnet – Project IST-2000-26434. Original URL is not operative, copy can be accessed at <http://crowley-coutaz.fr/FGnet/html/home.html>.
- [13] R. Diaz and A. Marathe. Soft labels for ordinal regression. In: *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4733–4742. Long Beach, CA, USA, 15–20 Jun 2019. doi:10.1109/CVPR.2019.00487.

- [14] M. Y. E. Dib and H. M. Onsi. Human age estimation framework using different facial parts. *Egyptian Informatics Journal*, 12(1):53–59, 2011. doi:10.1016/j.eij.2011.02.002.
- [15] F. Dornaika, I. Arganda-Carreras, and C. Belver. Age estimation in facial images through transfer learning. *Machine Vision and Applications*, 30(1):177–187, 2019. doi:10.1007/s00138-018-0976-1.
- [16] M. Duan, K. Li, and K. Li. An ensemble cnn2elm for age estimation. *IEEE Transactions on Information Forensics and Security*, 13(3):758–772, 2018. doi:10.1109/TIFS.2017.2766583.
- [17] E. Eidinger, R. Enbar, and T. Hassner. Age and gender estimation of unfiltered faces. *IEEE Transactions on Information Forensics and Security*, 9(12):2170–2179, 2014. doi:10.1109/TIFS.2014.2359646.
- [18] Y. Fu, G. Guo, and T. S. Huang. Age synthesis and estimation via faces: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(11):1955–1976, 2010. doi:10.1109/TPAMI.2010.36.
- [19] X. Geng, Z.-H. Zhou, Y. Zhang, G. Li, and H. Dai. Learning from facial aging patterns for automatic age estimation. In: K. Nahrstedt, M. Turk, Y. Rui, W. Klas, and K. Mayer-Patel, eds., *Proc. MM '06: Proc. 14th ACM International Conference on Multimedia*, pp. 307–316. ACM, Santa Barbara, CA USA, 23-27 Oct 2006. doi:10.1145/1180639.1180711.
- [20] P. A. George and G. J. Hole. Factors influencing the accuracy of age-estimates of unfamiliar faces. *Perception*, 24(9):1059–1073, 1995. doi:10.1068/p241059.
- [21] G. Guo, G. Mu, Y. Fu, and T. S. Huang. Human age estimation using bio-inspired features. In: *Proc. 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2009*, pp. 112–119. Miami, FL, USA, 20-25 Jun 2009. doi:10.1109/CVPRW.2009.5206681.
- [22] T. Hassner. The OUI-Adinece Face Image project. The Open University of Israel. <https://talhassner.github.io/home/projects/Adience/Adience-data.html>, [Accessed May 2024].
- [23] C. Kong, Q. Luo, and G. Chen. Learning deep contrastive network for facial age estimation. In: *Proc. International Joint Conference on Neural Networks (IJCNN)*. IEEE, Padua, Italy, 18-23 Jul 2022. doi:10.1109/IJCNN55064.2022.9892308.
- [24] Y. H. Kwon and N. da Vitoria Lobo. Age classification from facial images. In: *Proc. IEEE International Conference on Computer Vision and Pattern Recognition (ICCVPR)*, p. 762–767. Seattle, WA, USA, 21-23 Jun 1994. doi:10.1109/CVPR.1994.323894.
- [25] A. Lanitis. On the significance of different facial parts for automatic age estimation. In: *Proc. International Conference on Digital Signal Processing (DSP)*, vol. 2, pp. 1027–1030. Santorini, Greece, 01-03 Jul 2002. doi:10.1109/ICDSP.2002.1028265.
- [26] G. Levi and T. Hassner. Age and gender classification using convolutional neural networks. In: *Proc. 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 34–42. Boston, MA, USA, 07-12 Jun 2015. doi:10.1109/CVPRW.2015.7301352.
- [27] W. Li, J. Lu, J. Feng, C. Xu, J. Zhou, et al. BridgeNet: A continuity-aware probabilistic network for age estimation. In: *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1145–1154. Long Beach, CA, USA, 15-20 Jun 2019. doi:10.1109/CVPR.2019.00124.
- [28] H. Liu, J. Lu, J. Feng, and J. Zhou. Ordinal deep feature learning for facial age estimation. In: *Proc. 12th IEEE International Conference on Automatic Face and Gesture Recognition, (FG)*, pp. 157–164, 30 May – 03 Jun 2017. doi:10.1109/FG.2017.28.
- [29] H. Liu, J. Lu, J. Feng, and J. Zhou. Ordinal deep learning for facial age estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(2):486–501, 2019. doi:10.1109/TCSVT.2017.2782709.

- [30] K. H. Liu, S. Yan, and C.-C. J. Kuo. Age estimation via grouping and decision fusion. *IEEE Transactions on Information Forensics and Security*, 10(11):2408–2423, 2015. doi:10.1109/TIFS.2015.2462732.
- [31] Z. Lou, F. Alnajar, J. M. Alvarez, N. Hu, and T. Gevers. Expression-invariant age estimation using structured learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(2):365–375, 2018. doi:10.1109/TPAMI.2017.2679739.
- [32] M. Minear and D. C. Park. A lifespan database of adult facial stimuli. *Behavior Research Methods, Instruments, and Computers*, 36(4):630–633, 2004. doi:10.3758/BF03206543.
- [33] S. H. Nam, Y. H. Kim, N. Q. Truong, J. Choi, and K. R. Park. Age estimation by super-resolution reconstruction based on adversarial networks. *IEEE Access*, 8:17103–17120, 2020. doi:10.1109/ACCESS.2020.2967800.
- [34] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002. doi:10.1109/TPAMI.2002.1017623.
- [35] H. Pan, H. Han, S. Shan, and X. Chen. Mean-variance loss for deep age estimation from a face. In: *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5285–5294. Salt Lake City, UT, USA, 18–23 Jun 2018. doi:10.1109/CVPR.2018.00554.
- [36] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In: *Proc. 26th British Machine Vision Conference (BMVC)*, pp. 41.1–41.12. Swansea, UK, 7–10 Sep 2015. (doi inoperative). doi:10.5244/c.29.41, <https://bmva-archive.org.uk/bmvc/2015/papers/paper041/>.
- [37] R. Ranjan, S. Sankaranarayanan, C. D. Castillo, and R. Chellappa. An all-in-one convolutional neural network for face analysis. In: *Proc. 12th IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, pp. 17–24. Washington, DC, USA, 30 May – 03 Jun 2017. doi:10.1109/FG.2017.137.
- [38] G. Rhodes. Lateralized processes in face recognition. *British Journal of Psychology*, 76(2):249–271, 1985. doi:10.1111/j.2044-8295.1985.tb01949.x.
- [39] P. Rodríguez, G. Cucurull, J. M. Gonfaus, F. X. Roca, and J. González. Age and gender recognition in the wild with deep attention. *Pattern Recognition*, 72:563–571, 2017. doi:10.1016/j.patcog.2017.06.028.
- [40] R. Rothe, R. Timofte, and L. V. Gool. Deep expectation of real and apparent age from a single image without facial landmarks. *International Journal of Computer Vision*, 126(2):144–157, 2018. doi:10.1007/s11263-016-0940-3.
- [41] W. Samek, A. Binder, S. Lapuschkin, and K.-R. Müller. Understanding and comparing deep neural networks for age and gender classification. In: *Proc. 2017 IEEE International Conference on Computer Vision Workshops, (ICCVW)*, pp. 1629–1638. Venice, Italy, 22–29 Oct 2017. doi:10.1109/ICCVW.2017.191.
- [42] W. Shen, Y. Guo, Y. Wang, K. Zhao, B. Wang, et al. Deep regression forests for age estimation. In: *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2304–2313. Salt Lake City, UT, USA, 18–23 Jun 2018. doi:10.1109/CVPR.2018.00245.
- [43] W. Shen, Y. Guo, Y. Wang, K. Zhao, B. Wang, et al. Deep differentiable random forests for age estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(2):404–419, 2019. doi:10.1109/tpami.2019.2937294.
- [44] C. Shi, S. Zhao, K. Zhang, Y. Wang, and L. Liang. Face-based age estimation using improved swin transformer with attention-based convolution. *Frontiers in Neuroscience*, 17, 2023. doi:10.3389/fnins.2023.1136934.

- [45] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In: *Proc. 3rd International Conference on Learning Representations (ICLR)*. San Diego, CA, USA, 7-9 May 2015. Published only on arXiv. <http://arxiv.org/abs/1409.1556>.
- [46] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In: *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2818–2826. Las Vegas, NV, USA, 27-30 Jun 2016. doi:10.1109/CVPR.2016.308.
- [47] Z. Tan, J. Wan, Z. Lei, R. Zhi, G. Guo, et al. Efficient group-n encoding and decoding for facial age estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(11):2610–2623, 2018. doi:10.1109/TPAMI.2017.2779808.
- [48] J. C. Xie and C. M. Pun. Deep and ordinal ensemble learning for human age estimation from facial images. *IEEE Transactions on Information Forensics and Security*, 15(8):2361–2374, 2020. doi:10.1109/TIFS.2020.2965298.
- [49] H.-F. Yang, B.-Y. Lin, K.-Y. Chang, and C.-S. Chen. Automatic age estimation from face images via deep ranking. In: *Proc. 26th British Machine Vision Conference (BMVC)*, pp. 55.1–55.11. Swansea, UK, 7-10 Sep 2015. (doi inoperative). doi:10.5244/C.29.55, <https://bmva-archive.org.uk/bmvc/2015/papers/paper055/>.
- [50] C. Zhang and G. Guo. Age estimation with expression changes using multiple aging subspaces. In: *Proc. IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pp. 1–6. Arlington, VA, USA, 29 Sep – 02 Oct 2013. doi:10.1109/BTAS.2013.6712720.
- [51] C. Zhang, S. Liu, X. Xu, and C. Zhu. C3AE: Exploring the limits of compact model for age estimation. In: *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12579–12588. Long Beach, CA, USA, 15-20 Jun 2019. doi:10.1109/CVPR.2019.01287.

Joseph D. Akinyemi is currently with the University of York, York, United Kingdom. He received his Bachelor's degree in Computer Science from the University of Ilorin, Ilorin, Nigeria in 2010. He received his Master's degree in Computer Science from the University of Ibadan, Ibadan, Nigeria, in 2014 and a Ph.D. degree in Computer Science from the same institution in 2020. His research spans areas of Computer Vision such as facial and medical image processing as well as aspects of Natural Language Processing such as Sentiment Analysis. He is a 2022 Heidelberg Laureate Forum Fellow in Germany, a recipient of the Google Developers Machine Learning Bootcamp sponsorship for Sub-Saharan Africa and a member of the ACM.

Olufade F. W. Onifade is currently Professor of Computer Science at the University of Ibadan, Ibadan, Nigeria and a Deputy Director at the Open and Distance Learning Center of the University of Ibadan, Ibadan, Nigeria. He received his Bachelors degree in Mathematics and Computer Science at the Federal University of Agriculture, Abeokuta, Nigeria in 1998 and received his Masters degree at the University of Ibadan, Ibadan, Nigeria in 2002. He benefitted from the French government scholarship which led him to receive double doctorate degrees one from the University of Ibadan, Ibadan, Nigeria, and the other from Nancy 2 University, France, in 2010. His research interests are in Information Retrieval, Risk Management, Pattern Recognition and Computer Vision. He is a member of the IEEE, IAENG, ISKO and NCS. He has received a number of grants and awards including the MIT-ETT fellowship for Content Development and Delivery and the CV Raman Fellowship for African researchers in India. He is a well-cited author of over 80 papers in peer-reviewed journals and conferences.

