

Supplementary Materials: "Multi-modal Salient Object Detection via a Unified Diffusion Model"

In this supplementary material, we provide additional implementation details and experimental results. Our code is publicly available at: <https://github.com/QuantumScriptHub/diffSOD>.

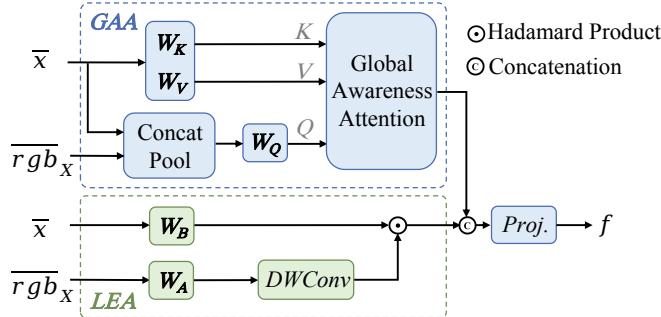


Figure 1: Diagrammatic details on how to conduct interactions between RGB and depth features (or thermal features). It contains three parts, which are responsible for global, local details, and extract surface information respectively. The base module is omitted for simplification.

1 IMPLEMENTATION DETAILS

Multi-resolution noise. Prior research has investigated variations from the classical DDPM approaches, introducing elements like atypical noise distributions or shortcuts that diverge from the Markovian sequence. The framework we introduce, along with the detailed fine-tuning procedure, accommodates alterations to the noise sequence during the fine-tuning phase. We discovered that a synergistic blend of noise applied at multiple resolutions, coupled with a gradually decreasing schedule, accelerates convergence and significantly enhances outcomes compared to the conventional DDPM approach. This layered noise approach involves layering a series of Gaussian noise patterns at varying scales, each scaled up to match the resolution required for the input of U-Net. Our suggested graduated schedule smoothly transitions from this layered, multi-resolution noise at $t = T$ to the typical Gaussian noise at $t = 0$. Furthermore, the gradual reduction of multi-resolution noise contributes to an additional improvement. We have also observed that training with multi-resolution noise results in more consistent predictions, even when different initial noise is used during inference. The annealing process further enhances this consistency.

Test-time ensembling. The probabilistic nature of the inference pipeline results in diverse predictions, contingent upon the initialization noise present in \mathbf{z}_T^y . Leveraging this, we introduce a test-time ensemble approach capable of amalgamating numerous inference iterations on the same input. For every input sample, there is the option to perform inference up to N times. This approach provides a versatile balance between computational efficiency and predictive accuracy, determined by the chosen value of N . We have observed a consistent improvement as the ensemble size increases. However,

this improvement begins to diminish after reaching 10 predictions per sample.

Denoising steps. We assess the impact of re-spaced inference denoising steps guided by the DDIM scheduler. Despite being trained with 1000 DDPM steps, opting for 50 steps proves adequate for accurate inference results. As anticipated, employing more denoising steps leads to superior outcomes. We note that the point of diminishing returns with additional denoising steps varies across datasets but consistently remains below 10 steps. This suggests that denoising steps can be further reduced to 10 or fewer for increased efficiency while maintaining comparable performance. Interestingly, this threshold is lower than the typical requirement of 50 steps for diffusion-based image generators.

2 ATTENTION FEATURE INTERACTION MODULE

Building Block. Our building attention feature interaction module (AFIM) is mainly composed of the global awareness attention (GAA) module and the local enhancement attention (LEA) module and builds interaction between the RGB and depth modalities (or thermal modalities). GAA incorporates depth information (or thermal information) and aims to enhance the capability of object localization from a global perspective, while LEA adopts a large-kernel convolution to capture the local clues from the depth features (or thermal features), which can refine the details of the RGB representations. The AFIM is an attention mechanism, leveraging depth information (or thermal information) to enhance the 3D awareness of RGB features, for better global context understanding, while the latter uses depth features (or thermal features) to augment the geometry details of RGB ones by a Hadamard product. The details of the interaction modules are shown in Fig. 1. Besides, we also preserve a base operation that only process RGB features to extract the pure color cues. Different from the heavy cross-attention mechanisms that present quadratic relationship to the number of input pixels or tokens, our block can efficiently use the depth or thermal cues for scene understanding and shape awareness. This makes our encoder able to interact RGB and depth or thermal cues densely. Hadamard product to aggregate global context information from the feature maps, where the first one leverages features extracted from depth (or thermal) and RGB as attention to help semantic discrimination and the other utilize the geometry shape information in depth feature (or thermal feature) to refine the RGB feature towards better awareness of object shapes. Instead of using heavy attention operations, where the computation cost present quadratic growth as the pixels or tokens increase, we implement the attention mechanism under a small fixed size and the enhancement for 3D geometry via Hadamard product, which are efficient yet effective. Thus, we can apply the interaction between RGB and depth (or thermal) in each blocks without heavy burdens.

Our GAA fuses depth (or thermal) and RGB features to build relationships across the whole scene, enhancing 3D awareness

and further helping capture semantic objects. Different from the self-attention mechanism that introduces quadratic computation growth as the pixels or tokens increase, the Query (Q) in GAA is down-sampled to a fixed size and hence the computational complexity can be reduced. It illustrates that fusing depth features(or thermal features) with Q is adequate and there is no need to combine them with K or V , which brings computation increment but no performance improvement. And we implement this interaction in a fixed small size. So, Q comes from the concatenation of the RGB features and depth features (or thermal features), while key (K) and value (V) are extracted from RGB features. Given the RGB features \bar{x}_i and depth features \overline{rgb}_{X_i} , the above process can be formulated as:

$$Q = \text{Linear}(\text{Pool}_{k \times k}([\bar{x}_i, \overline{rgb}_{X_i}])), K = \text{Linear}(\bar{x}_i), V = \text{Linear}(\bar{x}_i), \quad (1)$$

where $[\cdot, \cdot]$ denotes the concatenation operation along the channel dimension, $\text{Pool}_{k \times k}(\cdot)$ performs adaptively average pooling operation across the spatial dimensions to $k \times k$ size, and Linear is linear transformation. Based on the generated $Q \in \mathbb{R}^{k \times k \times C^d}$, $K \in \mathbb{R}^{h \times w \times C^d}$, and $V \in \mathbb{R}^{h \times w \times C^d}$, where h and w are the height and width of features in the current stage, we formulate the GAA as follows:

$$X_{GAA} = \text{UP}(V \cdot \text{Softmax}(\frac{Q^T K}{\sqrt{C^d}})), \quad (2)$$

where $\text{UP}(\cdot)$ is a bilinear upsampling operation that converts the spatial size from $k \times k$ to $h \times w$. In practical use, Eq.2 can also be extended to a multi-head version, as done in the original self-attention, to augment the feature representations.

We also design the LEA module to capture more local details, which can be regarded as a supplement to the GAA module. Unlike most previous works that use addition and concatenation to fuse the RGB features and depth features(or thermal features). We conduct a depth-wise convolution with a large kernel on the depth features(or thermal features) and use the resulting features as attention weights to reweigh the RGB features via a simple Hadamard product . This is reasonable in that adjacent pixels with similar depth values (or thermal information) often belong to the same object and the 3D geometry information thereby can be easily embedded into the RGB features. To be specific, the calculation process of LEA can be defined as follows:

$$X_{LEA} = \text{DConv}_{k \times k}(\text{Linear}(\overline{rgb}_{X_i})) \odot \text{Linear}(\bar{x}_i), \quad (3)$$

where $\text{DConv}_{k \times k}$ is a depth-wise convolution with kernel size $k \times k$ and \odot is the Hadamard product.

To preserve the diverse appearance information, we also build a base module to transform the RGB features \bar{x}_i to x_{Base} , which has the same spatial size as X_{GAA} and X_{LEA} . The calculation process of x_{Base} can be defined as follows:

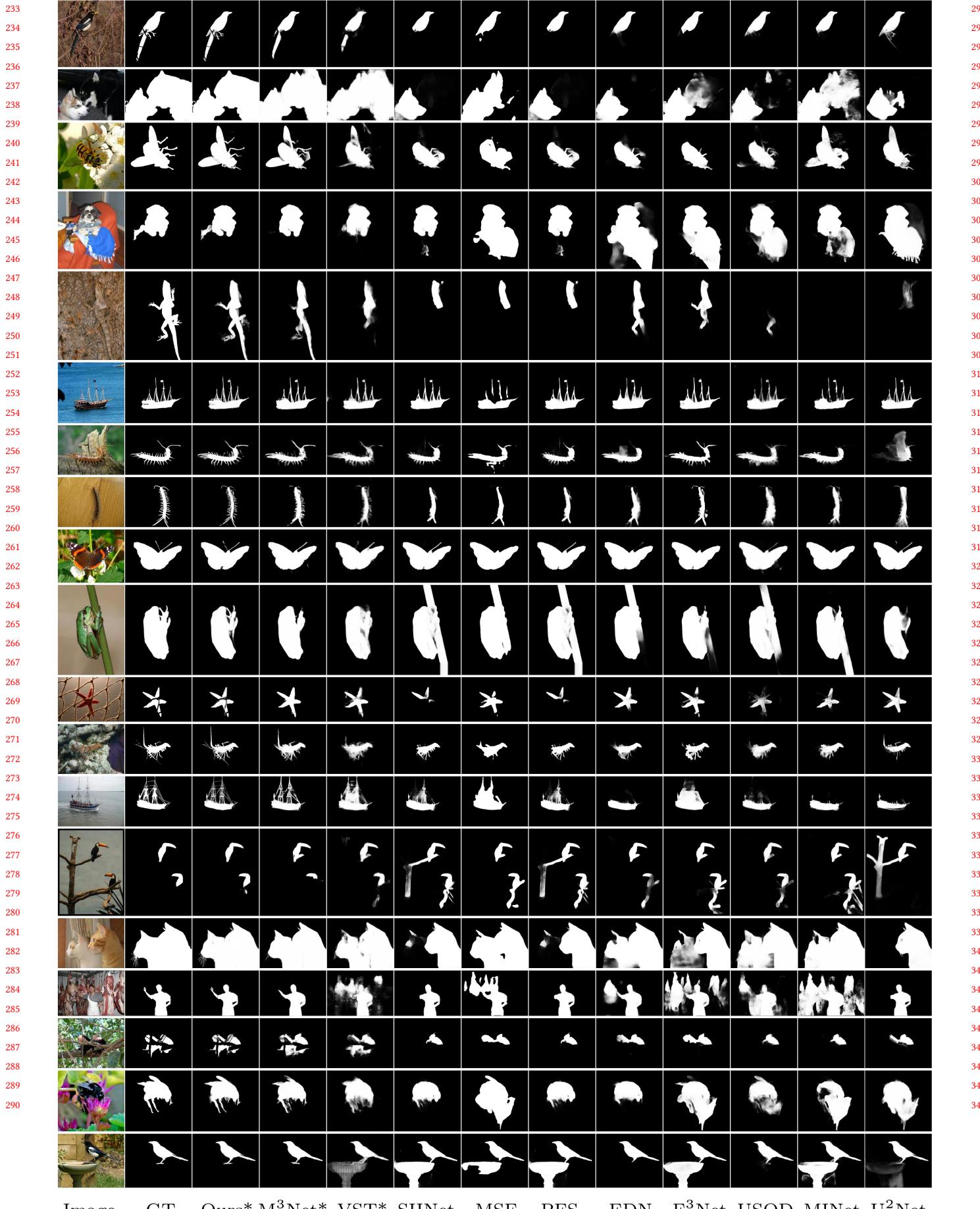
$$x_{Base} = \text{DConv}_{k \times k}(\text{Linear}(\bar{x}_i)) \odot \text{Linear}(\bar{x}_i). \quad (4)$$

Finally, the features, X_{GAA} , X_{LEA} , x_{Base} , are fused together by concatenation and linear projection to update the interaction features f .

3 EXPERIMENTAL RESULTS

From the results of methods based on non-transformer and transformers approaches presented in the paper, we can intuitively discern that the information from both global and local contexts is exceedingly beneficial for the integrity of salient object detection (SOD) predictions. Consequently, the state-of-the-art (SOTA) methods to date still predominantly rely on transformer-based approaches. However, existing methods continue to struggle with generating complete predictions, particularly in terms of edge refinement and fine details. This is primarily due to two factors: First, CNN-based encoders are not effective in capturing long-range dependencies for global context, leading to incomplete predictions. Second, downsampling the ground truth to fit the size of the prediction introduces inaccuracies, as details of the ground truth are lost during the interpolation or pooling process. We introduce Res2Nets as an auxiliary branch of our diffSOD encoder to better model long-term dependencies. The ablation experiments discussed in the main text regarding the addition of Res2Nets auxiliary branch also demonstrate that although our original diffusion model structure is already capable of yielding satisfactory results, the detection outcomes for salient targets can be further elevated by incorporating the Res2Nets auxiliary branch. As shown in Fig. 4, our network achieved the best results in precision and recall calculated across the benchmark datasets. It demonstrates outstanding performances of the proposed diffSOD. Due to space limitations, we presented the three different experiments together in the main text. In this supplementary material, we present them separately and provide additional experimental results with more comparative methods, offering a clearer demonstration of the superior performance of our approach. You can also use the provided code and pre-trained weights to perform inference on any RGB-D or RGB-T data pairs you have. If you encounter any issues, feel free to contact us at any time. We also provide more comparative experiments on RGB, RGB-D, and RGB-T. From a visual perspective, our method demonstrates significant advantages.

To provide more intuitive and equitable visual experimental results, we conducted inference on datasets unseen by the model. From these datasets, we randomly selected 150 images for display. For ease of layout, each display comprises three rows: the original image in the first row, the ground truth in the second row, and our model's prediction in the third row. We uniformly scaled each image, resulting in some appearing slightly distorted; however, upon enlargement, remarkable effects become evident. Evidently, our fine-tuned diffusion model leverages rich visual prior knowledge to impeccably capture features and edge information of salient targets. Remarkably, it even detects finer edge details than the ground truth, showcasing formidable zero-shot capabilities. Our diffSOD fully leverages the prior knowledge of salient objects in stable diffusion, and this fine-tuning strategy offers valuable insights for adapting diffusion models to other tasks. We have fully open-sourced both our training and inference code, allowing easy adaptation to other tasks with minimal input modifications. We hope to contribute even more to the open-source community through this effort.



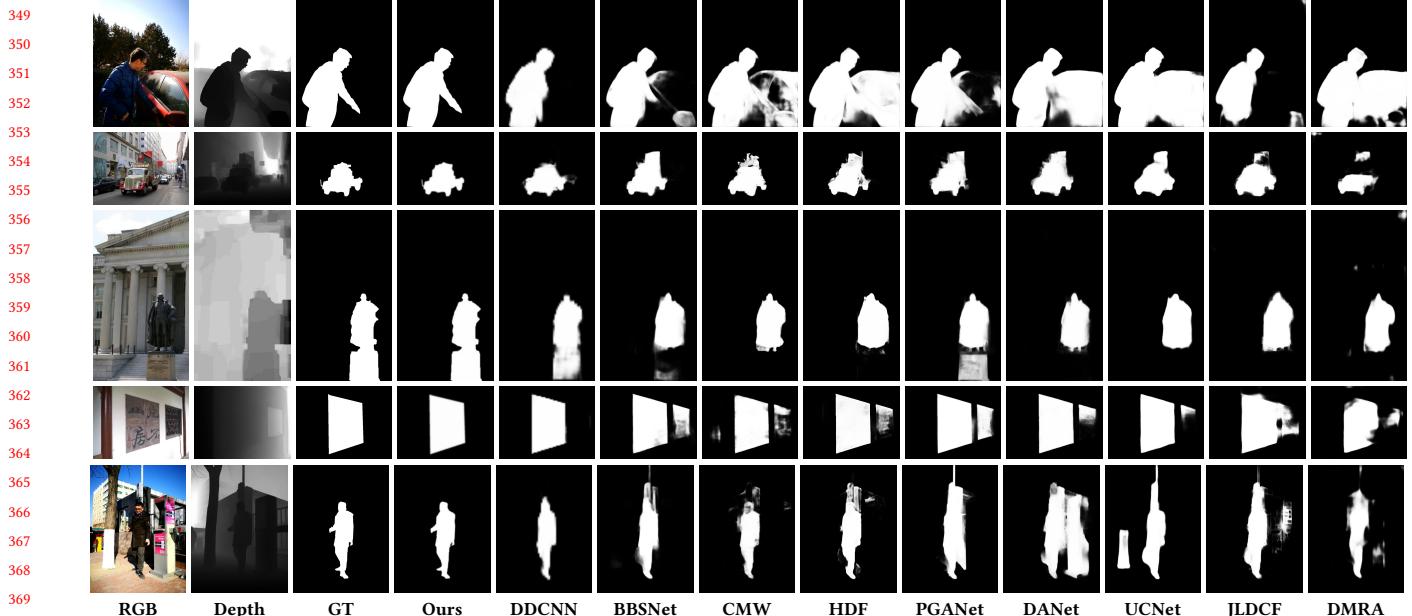


Figure 2: Visual comparison of saliency map results generated by various methods for RGB-D SOD.

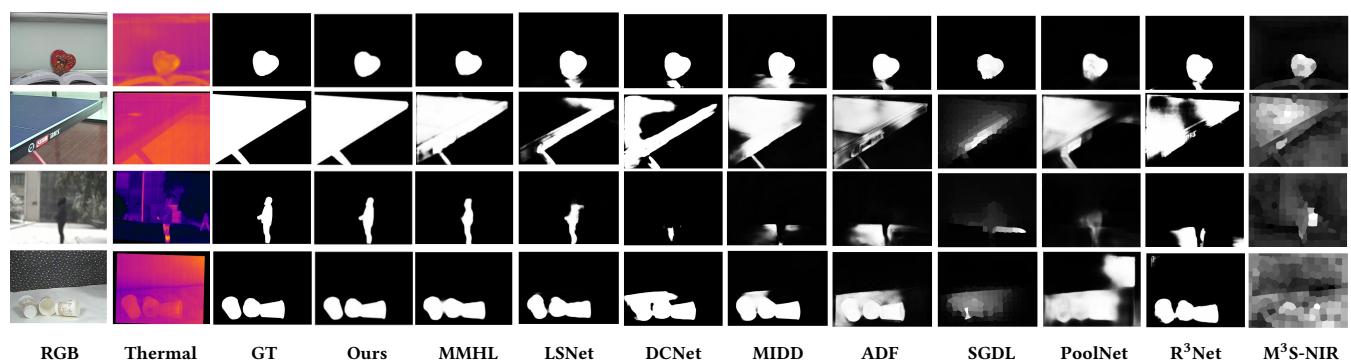
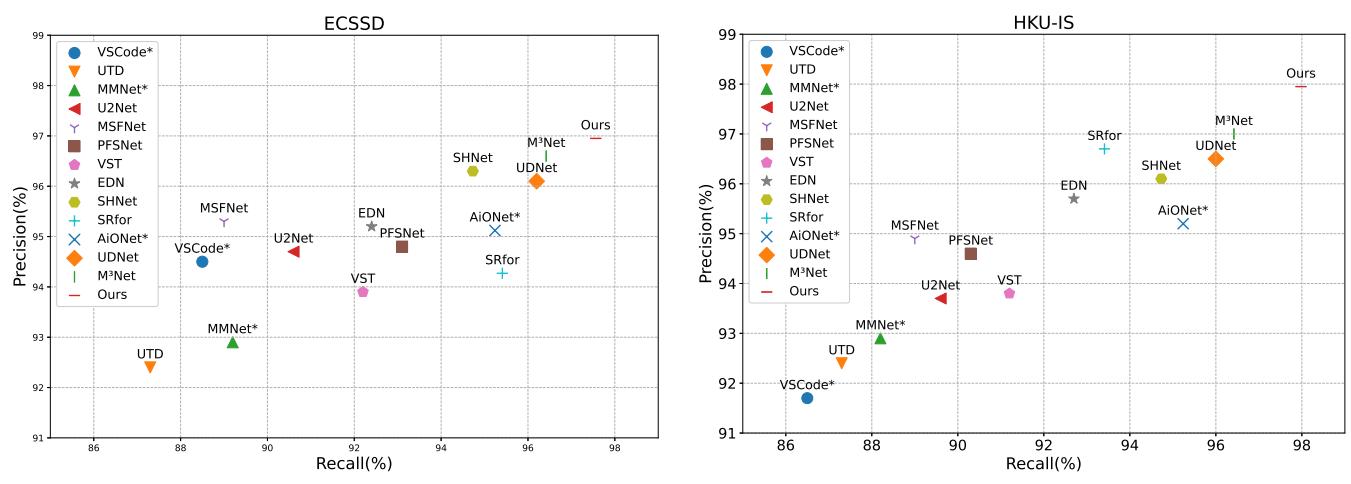


Figure 3: Visual comparison of saliency map results generated by various methods for RGB-T SOD.

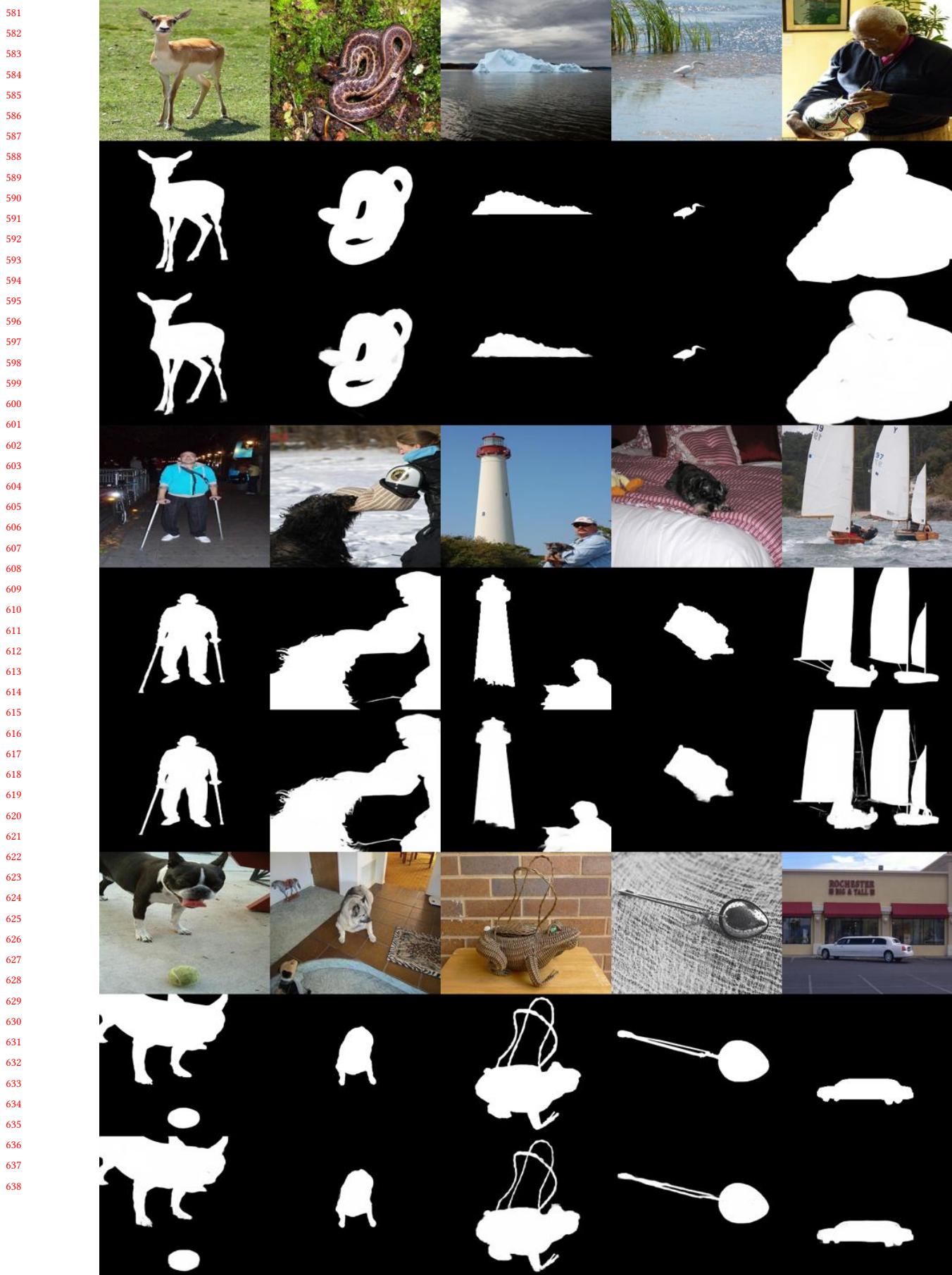


(a) The precision and recall on ECSSD.

(b) The precision and recall on HKU-IS.

Figure 4: The precision and recall of all the methods.

465						523
466						524
467						525
468						526
469						527
470						528
471						529
472						530
473						531
474						532
475						533
476						534
477						535
478						536
479						537
480						538
481						539
482						540
483						541
484						542
485						543
486						544
487						545
488						546
489						547
490						548
491						549
492						550
493						551
494						552
495						553
496						554
497						555
498						556
499						557
500						558
501						559
502						560
503						561
504						562
505						563
506						564
507						565
508						566
509						567
510						568
511						569
512						570
513						571
514						572
515						573
516						574
517						575
518						576
519						577
520						578
521						579
522						580



697



755

698

756

699

757

700

758

701

759

702

760

703

761

704

762

705

763

706

764

707

765

708

766

709

767

710

768

711

769

712

770

713

771

714

772

715

773

716

774

717

775



718

776

719

777

720

778

721

779

722

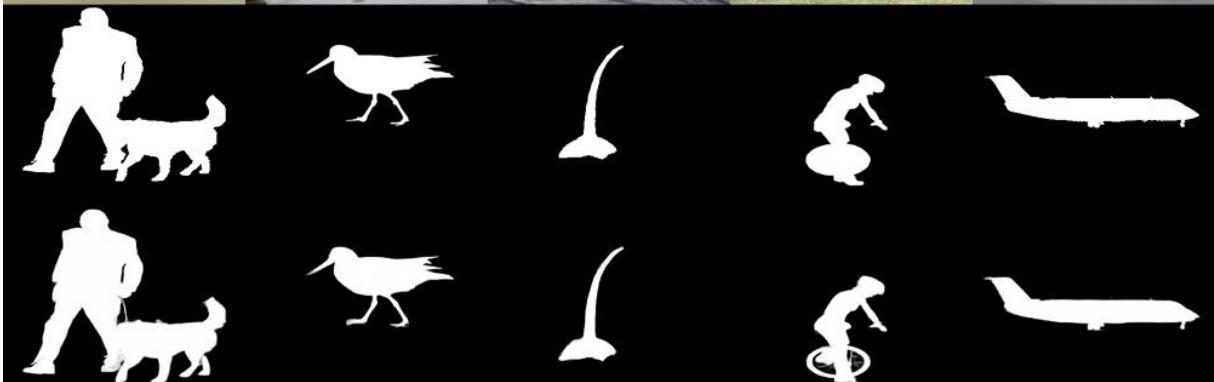
780

723

781

724

782



725

783

726

784

727

785

728

786

729

787

730

788

731

789

732

790

733

791

734

792

735

793

736

794

737

795



738

796

739

797

740

798

741

799

742

800

743

801

744

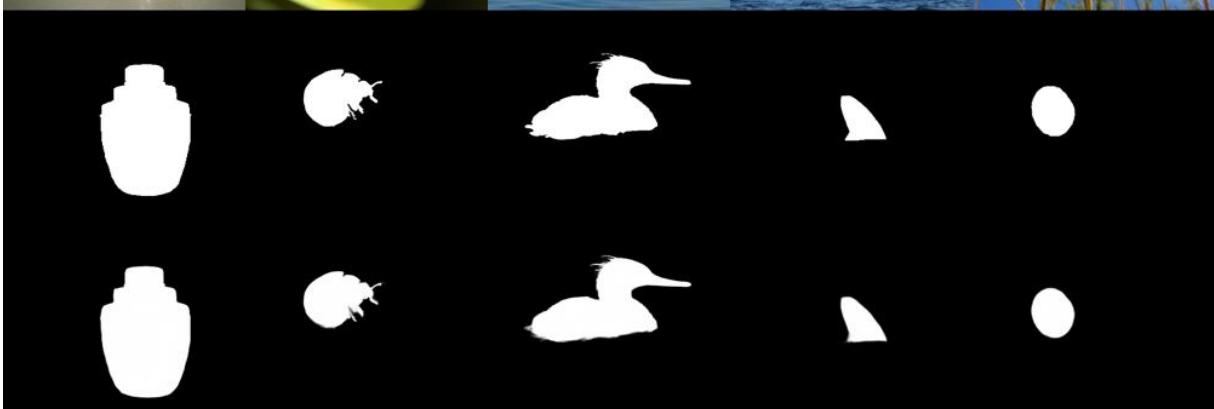
802

745

803

746

804



747

805

748

806

749

807

750

808

751

809

752

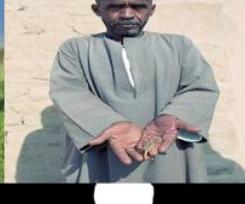
810

753

811

754

812

813						871
814						872
815						873
816						874
817						875
818						876
819						877
820						878
821						879
822						880
823						881
824						882
825						883
826						884
827						885
828						886
829						887
830						888
831						889
832						890
833						891
834						892
835						893
836						894
837						895
838						896
839						897
840						898
841						899
842						900
843						901
844						902
845						903
846						904
847						905
848						906
849						907
850						908
851						909
852						910
853						911
854						912
855						913
856						914
857						915
858						916
859						917
860						918
861						919
862						920
863						921
864						922
865						923
866						924
867						925
868						926
869						927
870						928

929						987
930						988
931						989
932						990
933						991
934						992
935						993
936						994
937						995
938						996
939						997
940						998
941						999
942						1000
943						1001
944						1002
945						1003
946						1004
947						1005
948						1006
949						1007
950						1008
951						1009
952						1010
953						1011
954						1012
955						1013
956						1014
957						1015
958						1016
959						1017
960						1018
961						1019
962						1020
963						1021
964						1022
965						1023
966						1024
967						1025
968						1026
969						1027
970						1028
971						1029
972						1030
973						1031
974						1032
975						1033
976						1034
977						1035
978						1036
979						1037
980						1038
981						1039
982						1040
983						1041
984						1042
985						1043
986						1044

1045



1046



1047



1048



1049



1050

1103

1104

1105

1106

1107

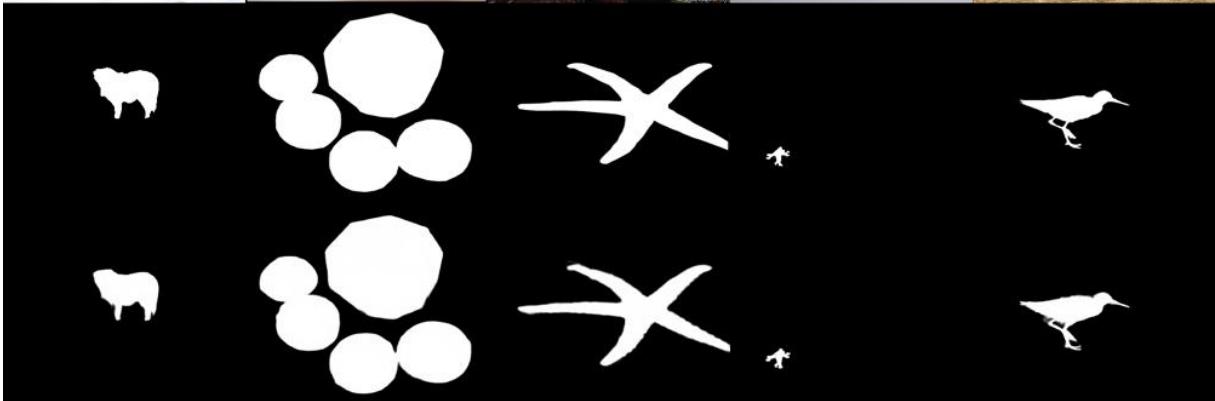
1108

1109

1051

1110

1052



1053

1111

1054

1112

1055

1113

1056

1114

1057

1115

1058

1116

1059

1117

1060

1118

1061

1119

1062

1120

1063

1121

1064

1122

1065

1123



1066

1124

1067

1125

1068

1126

1069

1127

1070

1128

1071

1129

1072

1130

1073

1131

1074

1132

1075

1133

1076

1134

1077

1135

1078

1136

1079

1137

1080

1138

1081

1139

1082

1140

1083

1141

1084

1142

1085

1143

1086

1144

1087

1145

1088

1146

1089

1147

1090

1148

1091

1149

1092

1150

1093

1151

1094

1152

1095

1153

1096

1154

1097

1155

1098

1156

1099

1157

1100

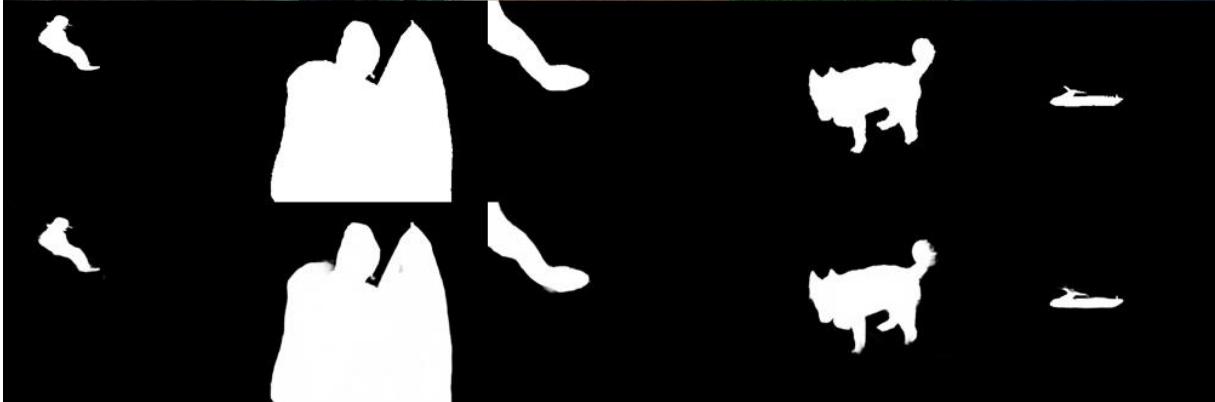
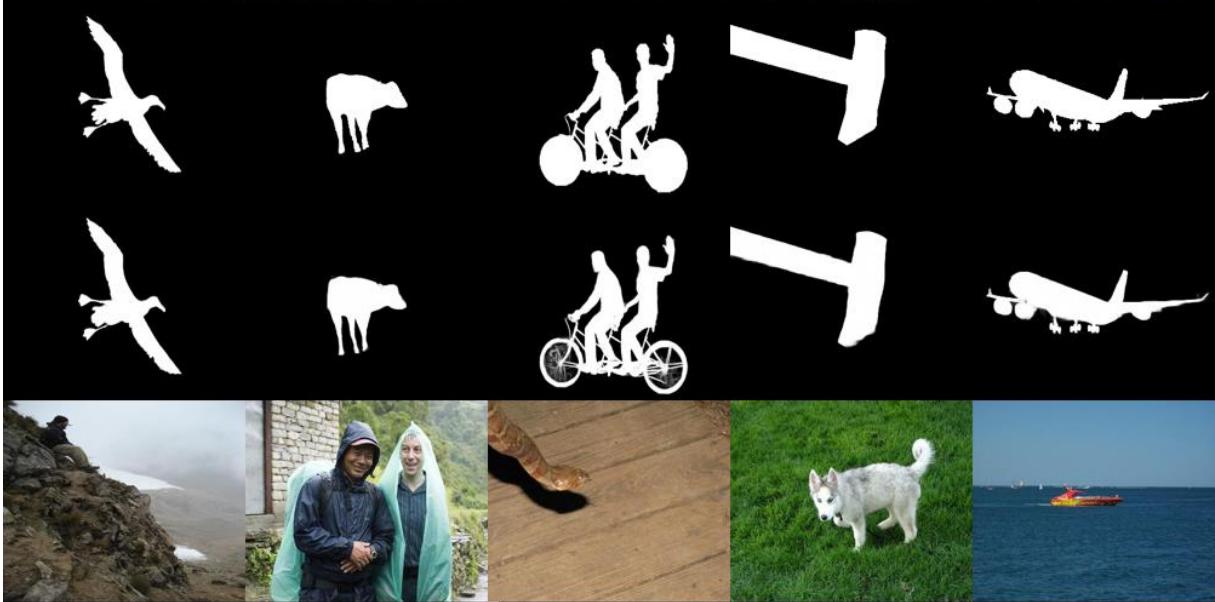
1158

1101

1159

1102

1160



1161



1219

1162

1220

1163

1221

1164

1222

1165

1223

1166

1224

1167

1225

1168

1226

1169

1227

1170

1228

1171

1229

1172

1230

1173

1231

1174

1232

1175

1233

1176

1234

1177

1235

1178

1236

1179

1237

1180

1238

1181

1239



1182

1240



1241

1183

1242

1184

1243

1185

1244

1186

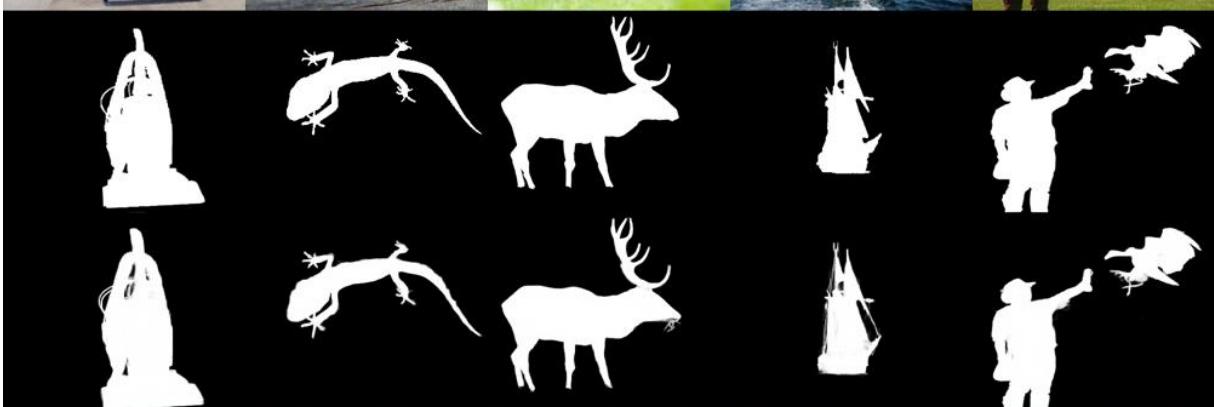
1245

1187

1246

1188

1247



1189

1248

1190

1249

1191

1250

1192

1251

1193

1252

1194

1253

1195

1254

1196

1255

1197

1256

1198

1257

1199

1258

1200

1259

1201

1260

1202

1261



1262

1203

1263

1204

1264

1205

1265

1206

1266

1207

1267

1208

1268

1209

1269

1210

1270

1211

1271

1212

1272

1213

1273

1214

1274

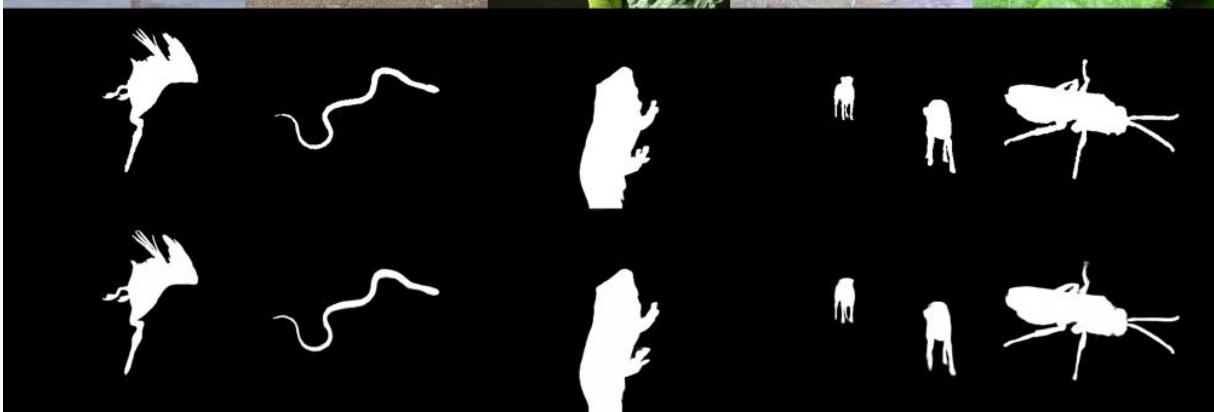
1215

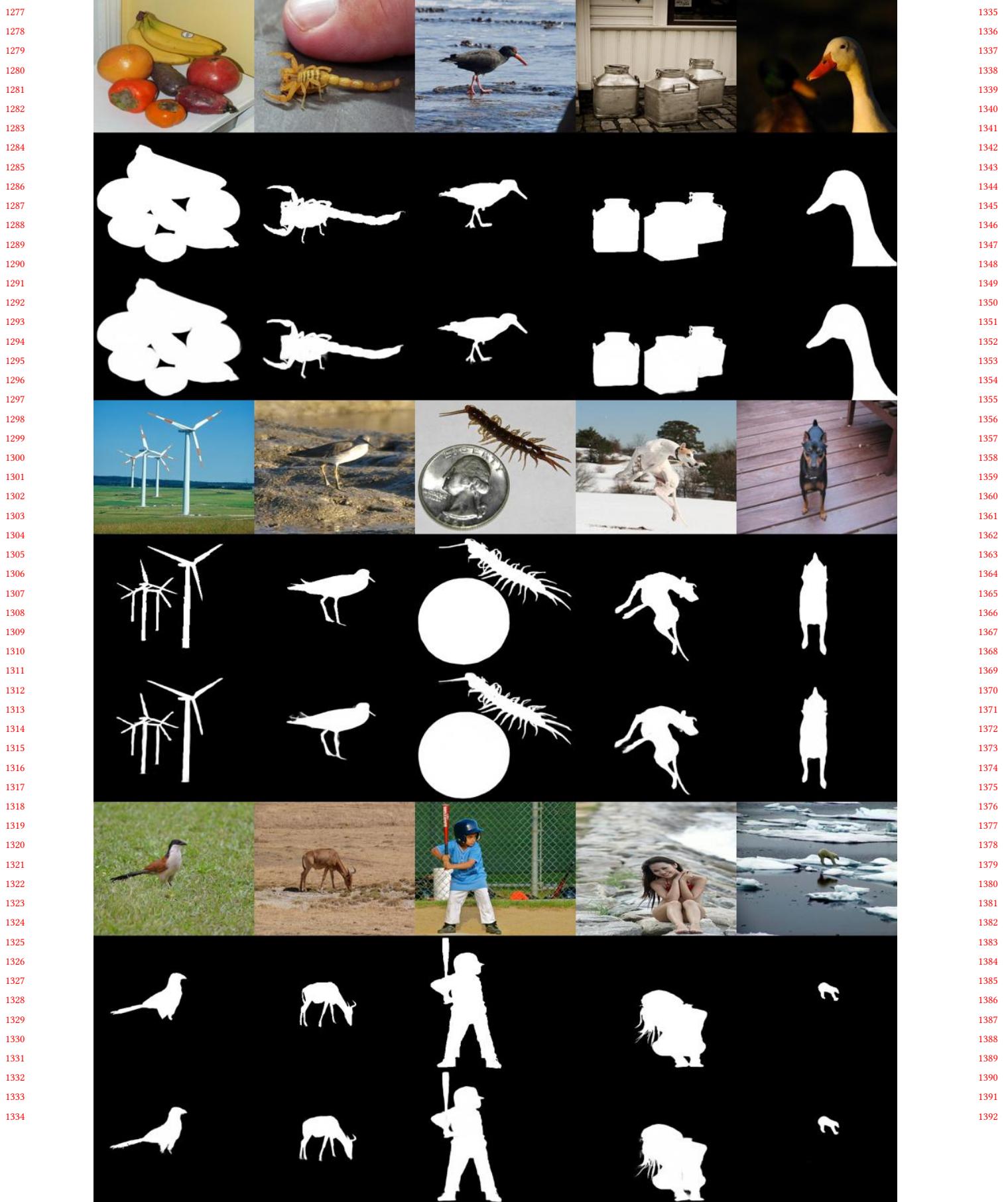
1275

1216

1276

1217





1393



1451

1394



1452

1395



1453

1396



1454

1397



1455

1398

1399



1456

1400



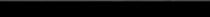
1457

1401



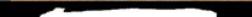
1458

1402



1459

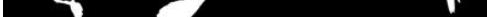
1403



1460

1404

1405



1461

1406

1462

1407



1463

1408



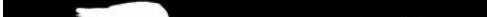
1464

1409



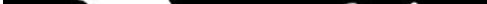
1465

1410



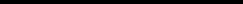
1466

1411



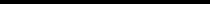
1467

1412

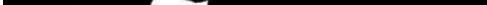


1468

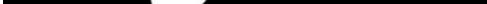
1413



1469

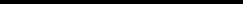


1470



1471

1414



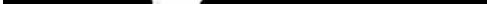
1472

1415



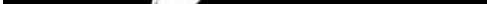
1473

1416



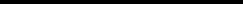
1474

1417



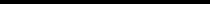
1475

1418



1476

1419



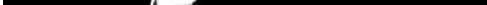
1477

1420

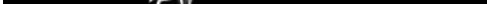


1478

1421

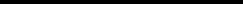


1479



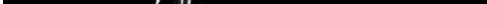
1480

1422



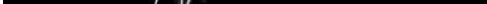
1481

1423



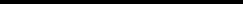
1482

1424



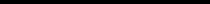
1483

1425



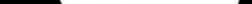
1484

1426



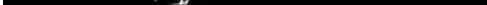
1485

1427



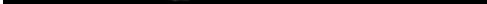
1486

1428



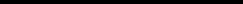
1487

1429



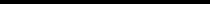
1488

1430



1489

1431



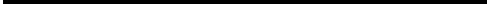
1490

1432



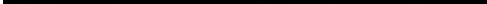
1491

1433



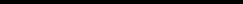
1492

1434



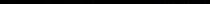
1493

1435



1494

1436



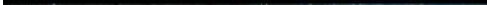
1495

1437



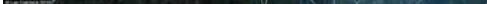
1496

1438



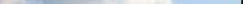
1497

1439



1498

1440

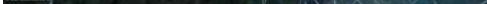


1499

1441



1500



1501



1502

1442



1503

1443



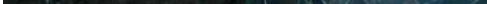
1504

1444



1505

1445



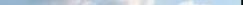
1506

1446



1507

1447



1508

1448



1509

1449

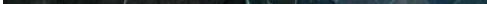


1510

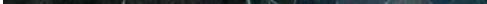
1450



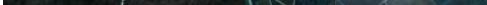
1511



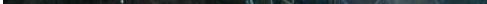
1512



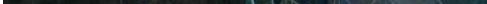
1513



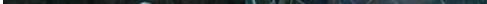
1514



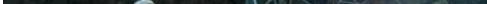
1515



1516



1517



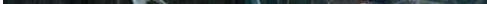
1518



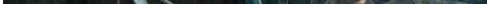
1519



1520



1521



1522



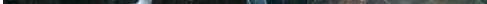
1523



1524



1525



1526



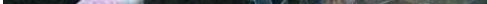
1527



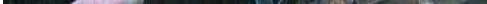
1528



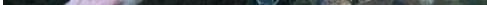
1529



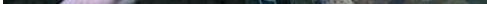
1530



1531



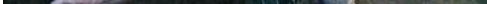
1532



1533



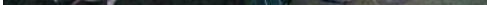
1534



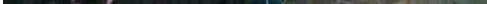
1535



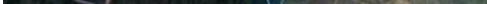
1536



1537



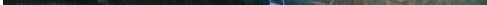
1538



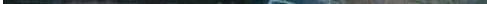
1539



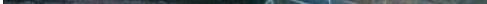
1540



1541



1542



1543



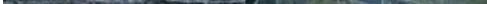
1544



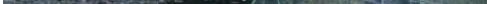
1545



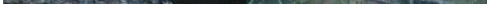
1546



1547



1548



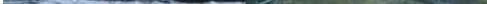
1549



1550



1551



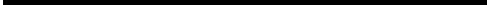
1552



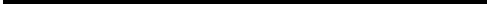
1553



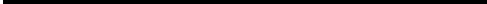
1554



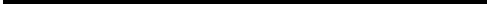
1555



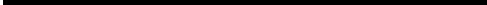
1556



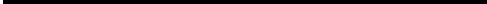
1557



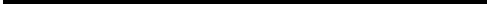
1558



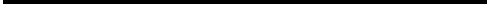
1559



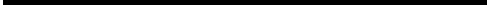
1560



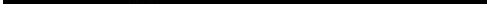
1561



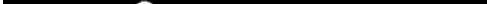
1562



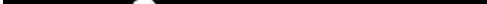
1563



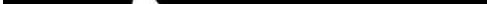
1564



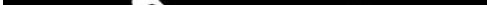
1565



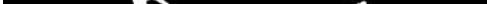
1566



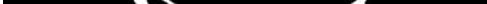
1567



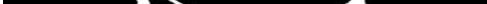
1568



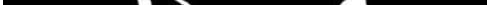
1569



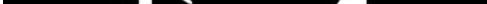
1570



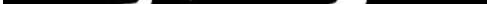
1571



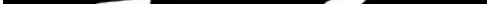
1572



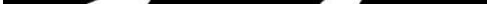
1573



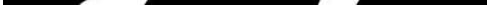
1574



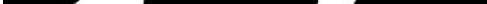
1575



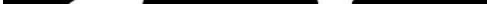
1576



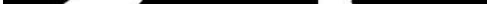
1577



1578



1579



1580

1509



1567

1510

1568

1511

1569

1512

1570

1513

1571

1514

1572

1515

1573

1516

1574

1517

1575

1518

1576

1519

1577

1520

1578

1521

1579

1522

1580

1523

1581

1524

1582

1525

1583

1526

1584

1527

1585

1528

1586

1529

1587



1588

1530

1589

1531

1590

1532

1591

1533

1592

1534

1593

1535

1594

1536

1595

1537

1596

1538

1597

1539

1598

1540

1599

1541

1600

1542

1601

1543

1602

1544

1603

1545

1604

1546

1605

1547

1606

1548

1607

1549

1608



1609

1550

1610

1551

1611

1552

1612

1553

1613

1554

1614

1555

1615

1556

1616

1557

1617

1558

1618

1559

1619

1560

1620

1561

1621

1562

1622

1563

1623

1564

1624

1565



1625